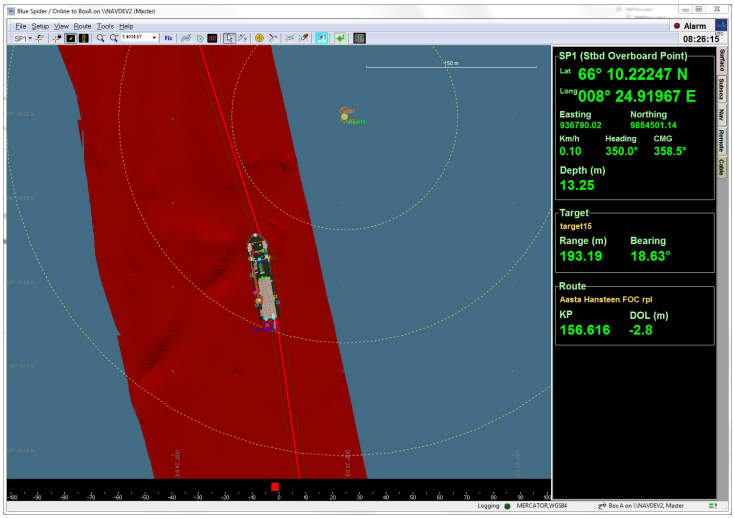
NAVSYSTEMS (IOM) LIMITED

Blue Spider

User Manual



2013

http://www.bluespider.im

The manual is issued in PDF format only

The manual is being regularly updated as new features are added to Blue Spider. Requests for any particular sections to be added should be emailed to support@bluespider.im

Release History	
~~Preliminary release~~	~~June 2011~~
~~1st Release~~	~~January 2012~~
2nd Release	December 2012
3rd Release	01/04/13
4th Release	01/06/13
5th Release	02/07/13

Table of Contents

1 Introduction.....26
2 Configuration.....27
2.1 Vessel Definition.....27
2.1.1 Info Tab.....31
2.1.1.1 Name.....31
2.1.1.2 MMSI Number. 31
2.1.1.3 Keel Height (Keel to CRP) 31
2.1.1.4 Draught (Keel to waterline).....31
2.1.1.5 Mass (ratio), Damping and Noise.....32
2.1.1.6 Minimum display size.....32
2.1.1.7 Save Definition and Load Definition.....32
2.1.2 Ref Points Tab.....33
2.1.2.1 Types of Offsets.....34
2.1.2.2 Pitch and Roll.....34
2.1.3 Outline Tab.....36
2.1.3.1 Saving and loading an outline.....37
2.1.4 Anchors Tab.....38
2.1.5 Display Tab.....40
2.1.5.1 Display Options.....40
2.1.5.2 Proximity Alert.....41
2.1.5.3 Appearance (Minimum Display Size).....43
2.1.6 Alignment Tab.....44
2.1.6.1 Position Offset.....45
2.1.6.2 Orientation.....45
2.1.6.3 Origin of rotation.....45
2.1.6.4 3D Appearance.....46
2.1.7 3D Files Tab.....47
2.1.7.1 Mesh files.....47
2.1.7.2 Material files.....48
2.1.7.3 Support files.....49

2.2 Mobile Definitions....50
2.2.1 Info Tab....51
2.2.1.1 Name....51
2.2.1.2 MMSI....51
2.2.1.3 Object Type....51
2.2.1.4 Mass....51
2.2.1.5 Appearance....51
2.2.2 Outline Tab....52
2.2.3 Points of Interest Tab....53
2.2.3.1 Beacon Names and Offsets....54
2.2.3.2 Depth Sensor Offset....54
2.2.4 Cable Detector Tab....55
2.2.4.1 How burial is calculated....57
2.2.5 Display Tab....59
2.2.6 Alignment Tab....60
2.2.7 3D Files Tab....60
2.3 Master / Slave....61
2.3.1 Master / Slave Messages....61
2.3.2 Master Slave Switch....62
2.3.3 Box A and Box B....63
2.3.4 DNS....65
2.3.5 Single Installation Box A (One Server)....66
2.3.5.1 Copy Config....67
2.3.5.2 Copy INI....68
2.3.5.3 Manually Copied Config Files....69
2.3.5.4 Machine.acl.ini....69
2.3.5.5 Navfix.ini....70
2.3.5.6 Nav-Inet.ini....70
2.3.5.7 PortManifest.cfg....70
2.3.6 Automatic Master / Slave....71
2.4 Data Communications....72
2.4.1 BSPNet....72
2.4.2 Configuring Ports....74
2.4.3 Editing Channel Names....74

2.5 Monitoring Ports....77
2.6 Decoding Alarms Message....78
2.6.1 Decoding Alarm Colours....80
2.7 Alert Logs....81
2.8 Steerpoints for Vessel and Mobiles....82
2.8.1 Vessel Steerpoints....82
2.8.2 Mobile Steerpoints....83
2.9 Mobile Configuration....84
2.10 Blue Spider and HPR Calibrations....85
2.10.1 The HPR System....85
2.11 Geodetics....87
2.12 The BSPEngine.ini File....89
2.12.1 BSPEngine.ini file Sections....90
2.13 Custom Data Inputs....91
2.13.1.1 String Formats....91
2.14 CSV Logging....95
2.15 SQL Logging....97
2.15.1 Brief Description....97
2.15.2 Advantages of SQL Logging....97
2.15.3 Limitations of SQL Logging....97
2.15.4 What gets recorded....98
3 Surveyor Tasks....99
3.1 Route Lines Points and Targets....99
3.1.1 Route Lines....99
3.1.2 Importing Routes....99
3.1.3 Importing Transformations....100
3.1.4 Viewing Route Lines, Points and Targets 101
3.1.4.1 Route Options....102
3.1.5 Creating Depths from Terrain Data....104
3.1.5.1 Maximum deviation from bathymetry....105
3.1.5.2 Sampling Step Size....105
3.1.6 Depth Charts....108
3.1.7 Creating Curved Routes....110
3.1.7.1 Route Track Properties....110

3.1.8 Creating Routes from Points....111
3.1.9 Creating Individual Radius Curves....112
3.1.10 Curve Radius Options....113
3.1.11 Grid, Altitude and Terrain Options....114
3.1.12 Targets....115
3.1.12.1 Create Targets....115
4 Barge Management System....118
4.1 Overview....118
4.2 Configuration for Barge Management....121
4.2.1 Equipment Required....121
4.2.1.1 Colubris....121
4.2.1.2 Radio Modem....122
4.2.2 Software Configuration....123
4.2.2.1 BSPEngine Configuration....123
4.2.2.2 Remote Vessel Configuration....124
4.2.2.3 Testing the Connection....125
4.2.3 Barge Management Features....126
4.2.3.1 Geodetics....126
4.2.3.2 Routes....126
4.2.3.3 AIS....126
4.2.3.4 Restricting permissions....126
4.2.3.5 Dropping own anchors....127
4.3 Barge Management System Operation....128
4.3.1 Introduction....128
4.3.2 Safety....129
4.3.3 Operation at Barge For Deployment....130
4.3.4 Operation at Tug for Deployment....135
4.3.5 Operation at Barge for Recovery....137
4.3.6 Anchor Recovery Operation at Tug....138
4.3.7 Anchor Racking....141
4.4 Anchor Logging....142
4.5 Anchor Winch Support....142
4.5.1 INI File Variables....143
4.5.1.1 BSPEngine Variables....143

5 Grid KP features....146
5.1 Recommendation....148
Appendix A....149
A.1 Introduction....150
A.1.1 Install location....150
A.1.2 Configuration data....151
A.1.2.1 Configuration location....151
A.1.2.2 System Alert Logs location....152
A.2 Variables....153
A.3 BSPEngine.INI....155
A.3.1 INI File Sections....155
A.3.1 INI File Reference....157
A.3.1.1 [System]....157
A.3.1.1.1 DefaultInputTimeout=....157
A.3.1.1.2 SpeedSmoothingInterval=....158
A.3.1.1.3 WaterlineSmoothingInterval=....158
A.3.1.1.4 MaxIntegratedNavTimeError=....158
A.3.1.1.5 MaxIntegratedNavDistanceError=....159
A.3.1.1.6 GPSAutoChangeoverDelay=....159
A.3.1.1.7 HPRPoleRotationCorrection=....160
A.3.1.1.8 AdjustHPRPitchAndRoll=....160
A.3.1.1.9 Rov1HPRAltitudeDisabled=....160
A.3.1.1.10 PloughHPRAltitudeDisabled=....160
A.3.1.1.11 PositionalSecrecy=....161
A.3.1.1.12 AllowCoordSysChanges=....161
A.3.1.1.13 AllowAnchorHandling=....161
A.3.1.1.14 MemoryUsageAlarmLimit....162
A.3.1.1.15 MemoryUsageSuicideLimit....162
A.3.1.1.16 StopSystemTimeGPSAdjustment....163
A.3.1.1.17 SQLServer....164
A.3.1.1.18 SQLPort....164
A.3.1.1.19 SQLUserName....164
A.3.1.1.20 SQLPassword....164
A.3.1.2 [Plc]...(Cable engine configuration)....165

A.3.1.3 [CustomInputFormat1]....166
A.3.1.3.1 MsgName=....167
A.3.1.3.2 MsgType=....167
A.3.1.3.3 Field1=....168
A.3.1.3.4 Delimiter=....168
A.3.1.3.5 Terminator=....168
A.3.1.3.6 Field specifiers....169
A.3.1.4 [CustomInputChannel1]....170
A.3.1.4.1 Message1=....171
A.3.1.5 [CustomOutputFormat1]....171
A.3.1.5.1 CustomOutputChannel=....173
A.3.1.5.2 LogToFile=....173
A.3.1.5.3 MsgName=....173
A.3.1.5.4 Field1= 173
A.3.1.5.5 Delimiter=....174
A.3.1.5.6 Terminator=....174
A.3.1.5.7 NMEA=checksum....174
A.3.1.5.8 WhenTimeout=....175
A.3.1.5.9 Trigger=....176
A.3.1.6 [Nav1]....178
A.3.1.7 [Gyro1]....181
A.3.1.8 [Depth1]....182
A.3.1.9 [RP01]...[RP03]....182
A.3.1.10 RTT Inputs [RTT_01]....183
A.3.1.10.1 radio_master/radio_slave=....184
A.3.1.10.2 remote_hpr=....185
A.3.1.10.3 grid_input=....186
A.3.1.10.4 pos_input=....186
A.3.1.10.5 multi_pos_input....187
A.3.1.10.6 All modes (except radio)....187
A.3.1.11 Examples....188
A.3.1.11.1 Phinns decode (pos_input)....188
A.3.1.11.2 Blueview (multi_pos_input)....190
A.3.1.11.3 Fanbeam (multi_pos_input)....190

A.3.1.11.4 Gps on a plough (pos_input).....191
A.3.1.12 [ScriptIncludes].....192
A.3.1.13 [Variables].....193
A.3.1.14 [VarHistory].....195
A.3.1.15 [LogFile1].....197
A.3.1.15.1 Title=.....199
A.3.1.15.2 Type=.....199
A.3.1.15.3 BaseFileName=.....202
A.3.1.15.4 Extension=.....202
A.3.1.15.5 DurationInHours=.....202
A.3.1.15.6 RateInSeconds=.....202
A.3.1.15.7 MaxFileSizeInBytes=.....203
A.3.1.15.8 MaxLines=.....203
A.3.1.15.9 Trigger=.....204
A.3.2 How to Decode Fields.....205
A.4 Machine.acl.INI.....206
A.5 Communications Device Names.....207
A.6 Built in Variables.....210
A.6.1 Variable Names.....211
A.6.1.1 AHT.Act.Date.....234
A.6.1.2 AHT.Act.Time.....235
A.6.1.3 AHT.Action.....235
A.6.1.4 AHT.Anchor.Name.....235
A.6.1.5 AHT.Anchor.Owner.....235
A.6.1.6 AHT.Drop.DeltaM.X.....236
A.6.1.7 AHT.Drop.DeltaM.Y.....236
A.6.1.8 AHT.Drop.GDelta.X.....236
A.6.1.9 AHT.Drop.GDelta.Y.....237
A.6.1.10 AHT.Drop.Grid.Easting.....237
A.6.1.11 AHT.Drop.Grid.Northing.....237
A.6.1.12 AHT.Drop.Pos.Lat.....238
A.6.1.13 AHT.Drop.Pos.Lon.....238
A.6.1.14 AHT.Grid.Easting.....238
A.6.1.15 AHT.Grid.Northing.....238

A.6.1.16 AHT.Pickup.DeltaM.X....239
A.6.1.17 AHT.Pickup.DeltaM.Y....239
A.6.1.18 AHT.Pickup.GDelta.X....239
A.6.1.19 AHT.Pickup.GDelta.Y....240
A.6.1.20 AHT.Pickup.Grid.Easting....240
A.6.1.21 AHT.Pickup.Grid.Northing....240
A.6.1.22 AHT.Pickup.Pos.Lat....240
A.6.1.23 AHT.Pickup.Pos.Lon....241
A.6.1.24 AHT.Pos.Lat....241
A.6.1.25 AHT.Pos.Lon....241
A.6.1.26 AHT.Requestor....241
A.6.1.27 AHT.Target.Grid.Easting....242
A.6.1.28 AHT.Target.Grid.Northing....242
A.6.1.29 AHT.Target.Pos.Lat....242
A.6.1.30 AHT.Target.Pos.Lon....242
A.6.1.31 AHT.Tug.Name....243
A.6.1.32 Alert.Description....243
A.6.1.33 AutoPilot.Direction....243
A.6.1.34 AutoPilot.ReversedDirection....243
A.6.1.35 AutoPilot.SP2.Direction....244
A.6.1.36 AutoPilot.SP2.ReversedDirection....244
A.6.1.37 Beacon.ID....244
A.6.1.38 Beacon.Pos.Lat....245
A.6.1.39 Beacon.Pos.Lon....245
A.6.1.40 Beacon.X....245
A.6.1.41 Beacon.Y....246
A.6.1.42 Beacon.Z....246
A.6.1.43 Cable.AUX1.Length....246
A.6.1.44 Cable.AUX1.SlackFromSectionStart...247
A.6.1.45 Cable.AUX1.Smoothed.Slack....247
A.6.1.46 Cable.AUX1.Smoothed.Speed....247
A.6.1.47 Cable.AUX1.Smoothed.Tension....247
A.6.1.48 Cable.AUX1.Speed....248
A.6.1.49 Cable.AUX1.Tension....248

A.6.1.50 Cable.AUX2.Length....248
A.6.1.51 Cable.AUX2.SlackFromSectionStart...248
A.6.1.52 Cable.AUX2.Smoothed.Slack....249
A.6.1.53 Cable.AUX2.Smoothed.Speed....249
A.6.1.54 Cable.AUX2.Smoothed.Tension....249
A.6.1.55 Cable.AUX2.Speed....249
A.6.1.56 Cable.AUX2.Tension....249
A.6.1.57 Cable.ControlSpeed....250
A.6.1.58 Cable.DistanceDeviation....250
A.6.1.59 Cable.Engine1.CableOut....250
A.6.1.60 Cable.Engine1.Tension....250
A.6.1.61 Cable.Engine2.CableOut....251
A.6.1.62 Cable.Engine2.Tension....251
A.6.1.63 Cable.Engine3.CableOut....251
A.6.1.64 Cable.Engine3.Tension....251
A.6.1.65 Cable.Engine4.CableOut....252
A.6.1.66 Cable.Engine4.Tension....252
A.6.1.67 Cable.Factory.Length....252
A.6.1.68 Cable.Grid.Easting....252
A.6.1.69 Cable.Grid.Northing....253
A.6.1.70 Cable.PLC1.Raw.Count....253
A.6.1.71 Cable.PLC1.Raw.Tension....253
A.6.1.72 Cable.PLC2.Raw.Count....254
A.6.1.73 Cable.PLC2.Raw.Tension....254
A.6.1.74 Cable.PLC3.Raw.Count....254
A.6.1.75 Cable.PLC3.Raw.Tension....254
A.6.1.76 Cable.Pos.Alt....255
A.6.1.77 Cable.Pos.Lat....255
A.6.1.78 Cable.Pos.Lon....255
A.6.1.79 Cable.Primary.Length....255
A.6.1.80 Cable.Primary.SlackFromSectionStart 256
A.6.1.81 Cable.Primary.Smoothed.Slack....256
A.6.1.82 Cable.Primary.Smoothed.Speed....256
A.6.1.83 Cable.Primary.Smoothed.Tension....256

A.6.1.84 Cable.Primary.Speed....257
A.6.1.85 Cable.Primary.Tension....257
A.6.1.86 Cable.RouteDistance....257
A.6.1.87 Cable.TargetSlack....257
A.6.1.88 Cable.TargetSpeedKmh....258
A.6.1.89 Cable.TargetTension....258
A.6.1.90 Clara.AutoSolveMode....258
A.6.1.91 Clara.CableInfo....258
A.6.1.92 Clara.MBTension....259
A.6.1.93 Clara.MSeabedSlope....259
A.6.1.94 Clara.UserAdjust....259
A.6.1.95 Clara.UseRouteDepth....260
A.6.1.96 Clara.UseRouteSlope....260
A.6.1.97 GPS1.Altitude....260
A.6.1.98 GPS1.AltitudeWGS84....260
A.6.1.99 GPS1.CRP.DX....261
A.6.1.100 GPS1.CRP.DY....261
A.6.1.101 GPS1.CRP.DZ....261
A.6.1.102 GPS1.CRP.Pos.Alt....261
A.6.1.103 GPS1.CRP.Pos.Lat....262
A.6.1.104 GPS1.CRP.Pos.Lon....262
A.6.1.105 GPS1.CRP.WGS84.Pos.Alt....262
A.6.1.106 GPS1.CRP.WGS84.Pos.Lat....262
A.6.1.107 GPS1.CRP.WGS84.Pos.Lon....263
A.6.1.108 GPS1.Date....263
A.6.1.109 GPS1.DatumShifted.Pos.Alt....263
A.6.1.110 GPS1.DatumShifted.Pos.Lat....263
A.6.1.111 GPS1.DatumShifted.Pos.Lon....264
A.6.1.112 GPS1.GeoidalSeparation....264
A.6.1.113 GPS1.GPS2.Heading....264
A.6.1.114 GPS1.Grid.Easting....264
A.6.1.115 GPS1.Grid.Northing....265
A.6.1.116 GPS1.HDOP....265
A.6.1.117 GPS1.PDOP....265

A.6.1.118 GPS1.Pos.Lat....265
A.6.1.119 GPS1.Pos.Lon....266
A.6.1.120 GPS1.Quality....266
A.6.1.121 GPS1.Sats....266
A.6.1.122 GPS1.Time....266
A.6.1.123 GPS1.VDOP....267
A.6.1.124 GPS2.GPS3.Heading....267
A.6.1.125 GPS3.GPS1.Heading....267
A.6.1.126 Gyro1.Corr.Heading....267
A.6.1.127 Gyro1.Heading....268
A.6.1.128 Gyro1.Message....268
A.6.1.129 Gyro1.Raw.Heading....268
A.6.1.130 HPR.Ancilliary.Heading....268
A.6.1.131 HPR.Ancilliary.Heave....269
A.6.1.132 HPR.Ancilliary.Pitch....269
A.6.1.133 HPR.Ancilliary.Roll....269
A.6.1.134 Logging.Backup1.AnticipatedSize....269
A.6.1.135 Logging.Backup1.FileSize....270
A.6.1.136 Logging.Backup1.Unc....270
A.6.1.137 Logging.Cable.Line.Name....270
A.6.1.138 Logging.Cable.Line.No....270
A.6.1.139 Logging.Cable.Type....271
A.6.1.140 Logging.Comment....271
A.6.1.141 Logging.Config1.LogType....271
A.6.1.142 Logging.Config1.Name....271
A.6.1.143 Logging.Description....272
A.6.1.144 Logging.EventNo....272
A.6.1.145 Logging.FixedSP....272
A.6.1.146 Logging.FixNo....272
A.6.1.147 Logging.Primary1.AnticipatedSize....273
A.6.1.148 Logging.Primary1.FileSize....273
A.6.1.149 Logging.Primary1.Unc....273
A.6.1.150 MRU1.Heave....273
A.6.1.151 MRU1.Pitch....274

A.6.1.152 MRU1.Roll.....274
A.6.1.153 Option.SpeedGaugeKmh.Max.....274
A.6.1.154 Option.SpeedGaugeKmh.Min.....274
A.6.1.155 Option.TensionGaugeKN.Max.....275
A.6.1.156 Option.TensionGaugeKN.Min.....275
A.6.1.157 PrimaryGPS.Altitude.....275
A.6.1.158 PrimaryGPS.AltitudeWGS84.....275
A.6.1.159 PrimaryGPS.GeoidalSeparation.....276
A.6.1.160 PrimaryGPS.HDOP.....276
A.6.1.161 PrimaryGPS.Quality.....276
A.6.1.162 PrimaryGPS.Sats.....277
A.6.1.163 Remotes.Vessel1.ID.....277
A.6.1.164 Remotes.Vessel1.Name.....277
A.6.1.165 Remotes.Vessel1.Push.Avg.Latency.277
A.6.1.166 Remotes.Vessel1.Push.Latency.....278
A.6.1.167 Remotes.Vessel1.SP1.Offset.Name..278
A.6.1.168 Remotes.Vessel1.SP1.Pos.Alt.....278
A.6.1.169 Remotes.Vessel1.SP1.Pos.Lat.....278
A.6.1.170 Remotes.Vessel1.SP1.Pos.Lon.....279
A.6.1.171 Remotes.Vessel1.Stats.Avg.TimeDelta
.....279
A.6.1.172 Remotes.Vessel1.System.Timestamp
.....279
A.6.1.173 Remotes.Vessel1.Time.Delta.....279
A.6.1.174
Remotes.Vessel1.Time.Estimated.Timestamp....280
A.6.1.175 Route.Direction.....280
A.6.1.176 Route.Name.....280
A.6.1.177 Route.Target.Name.....280
A.6.1.178 Route.Target1.Pos.Lat.....281
A.6.1.179 Route.Target1.Pos.Lon.....281
A.6.1.180 Route.Target1.WGS84.Pos.Lat.....281
A.6.1.181 Route.Target1.WGS84.Pos.Lon.....281
A.6.1.182 RTT_01.Altitude.....282

A.6.1.183 RTT_01.Heading....282
A.6.1.184 RTT_01.Pos.Lat....282
A.6.1.185 RTT_01.Pos.Lon....282
A.6.1.186 RTT_01.WaterDepth....283
A.6.1.187 Ship.AvgWaterLine....283
A.6.1.188 Ship.AvgWaterLineWGS84....283
A.6.1.189 Ship.CableEngines.PrimaryChannel..284
A.6.1.190 Ship.CRP.AltitudeWGS84....284
A.6.1.191 Ship.DesiredSpeedKmh....284
A.6.1.192 Ship.Draft....285
A.6.1.193 Ship.EchoSounderDepth....285
A.6.1.194 Ship.EchoSounderDepth1....285
A.6.1.195 Ship.GeoidWaterDepth....285
A.6.1.196 Ship.GeoidWaterDepth1....286
A.6.1.197 Ship.GPS.AltitudeWGS84....286
A.6.1.198 Ship.GPS.GeoidalSeparation....286
A.6.1.199 Ship.GPS.HDOP....286
A.6.1.200 Ship.GPS.Pos.Alt....287
A.6.1.201 Ship.GPS.Pos.Lat....287
A.6.1.202 Ship.GPS.Pos.Lon....287
A.6.1.203 Ship.GPS.Quality....287
A.6.1.204 Ship.GPS.ReceiverFlags....288
A.6.1.205 Ship.GPS.Sats....288
A.6.1.206 Ship.GPS.VTG.Course....288
A.6.1.207 Ship.GPS.VTG.Speed....289
A.6.1.208 Ship.GridHeading....289
A.6.1.209 Ship.Gyro.ReceiverFlags....289
A.6.1.210 Ship.Heading....290
A.6.1.211 Ship.Kalman.CMG....290
A.6.1.212 Ship.Kalman.Pos.Lat....290
A.6.1.213 Ship.Kalman.Pos.Lon....290
A.6.1.214 Ship.Kalman.Speed....291
A.6.1.215 Ship.KeelHeight....291
A.6.1.216 Ship.LaybackPoint....291

A.6.1.217 Ship.Motion.Heading.....291
A.6.1.218 Ship.Motion.Heave.....292
A.6.1.219 Ship.Motion.Pitch.....292
A.6.1.220 Ship.Motion.Roll.....292
A.6.1.221 Ship.MRU.ReceiverFlags.....292
A.6.1.222 Ship.Offsets.Grid1.Easting.....293
A.6.1.223 Ship.Offsets.Grid1.Northing.....293
A.6.1.224 Ship.Offsets.Pos1.Alt.....293
A.6.1.225 Ship.Offsets.Pos1.Elev.....294
A.6.1.226 Ship.Offsets.Pos1.Lat.....294
A.6.1.227 Ship.Offsets.Pos1.Lon.....294
A.6.1.228 Ship.Offsets.WGS84.Pos1.Alt.....294
A.6.1.229 Ship.Offsets.WGS84.Pos1.Lat.....295
A.6.1.230 Ship.Offsets.WGS84.Pos1.Lon.....295
A.6.1.231 Ship.PrimaryGyro.Message.....295
A.6.1.232 Ship.RawSpeedKmh.....295
A.6.1.233 Ship.SP1.Grid.Easting.....296
A.6.1.234 Ship.SP1.Grid.Northing.....296
A.6.1.235 Ship.SP1.Pos.Alt.....296
A.6.1.236 Ship.SP1.Pos.Elev.....296
A.6.1.237 Ship.SP1.Pos.Lat.....297
A.6.1.238 Ship.SP1.Pos.Lon.....297
A.6.1.239 Ship.SP1.Route.Arc.DOL.....297
A.6.1.240 Ship.SP1.Route.Arc.KP.....298
A.6.1.241 Ship.SP1.Route.DOL.....298
A.6.1.242 Ship.SP1.Route.Grid.DOL.....298
A.6.1.243 Ship.SP1.Route.Grid.KP.....298
A.6.1.244 Ship.SP1.Route.KP.....299
A.6.1.245 Ship.Speed.....299
A.6.1.246 Ship.SpeedKmh.....299
A.6.1.247 Ship.SpeedMS.....300
A.6.1.248 Ship.VDatumShift.....300
A.6.1.249 Ship.WaterDepth.....300
A.6.1.250 Ship.WaterDepth1.....301

A.6.1.251 Ship.WaterLine....301
A.6.1.252 Ship.WaterLineWGS84....301
A.6.1.253 SP1.Averaged.CMG....302
A.6.1.254 SP1.Averaged.Speed....302
A.6.1.255 SP1.Averaged.SpeedKmh....302
A.6.1.256 SP1.Date....302
A.6.1.257 SP1.GPS.AltitudeWGS84....303
A.6.1.258 SP1.GPS.GeoidalSeparation....303
A.6.1.259 SP1.GPS.HDOP....303
A.6.1.260 SP1.GPS.PDOP....304
A.6.1.261 SP1.GPS.Quality....304
A.6.1.262 SP1.GPS.Sats....304
A.6.1.263 SP1.GPS.VDOP....304
A.6.1.264 SP1.Grid.Easting....305
A.6.1.265 SP1.Grid.Northing....305
A.6.1.266 SP1.KP....305
A.6.1.267 SP1.Offset.Name....306
A.6.1.268 SP1.Pos.Alt....306
A.6.1.269 SP1.Pos.Lat....306
A.6.1.270 SP1.Pos.Lon....306
A.6.1.271 SP1.Route.DOL....307
A.6.1.272 SP1.Route.Grid.DOL....307
A.6.1.273 SP1.Route.Grid.KP....307
A.6.1.274 SP1.Route.KP....308
A.6.1.275 SP1.Route.SeabedSlope....308
A.6.1.276 SP1.Route.Section.Bearing....308
A.6.1.277 SP1.Route.Target.Bearing....308
A.6.1.278 SP1.Route.Target.Range....309
A.6.1.279 SP1.Route.TerrainDist....309
A.6.1.280 SP1.Route.WaterDepth....310
A.6.1.281 SP1.Smoothed.CMG....310
A.6.1.282 SP1.Speed....310
A.6.1.283 SP1.SpeedKmh....310
A.6.1.284 SP1.Target1.Bearing....311

A.6.1.285 SP1.Target1.Range....311
A.6.1.286 SP1.Time....311
A.6.1.287 SP1.WGS84.Pos.Alt....311
A.6.1.288 SP1.WGS84.Pos.Lat....312
A.6.1.289 SP1.WGS84.Pos.Lon....312
A.6.1.290 SP2.Grid.Easting....312
A.6.1.291 SP2.Grid.Northing....312
A.6.1.292 SP2.GridHeading....313
A.6.1.293 SP2.Heading....313
A.6.1.294 SP2.LaybackBearing....313
A.6.1.295 SP2.LaybackDistance....313
A.6.1.296 SP2.LaybackMode....314
A.6.1.297 SP2.Motion.Pitch....314
A.6.1.298 SP2.Motion.Roll....314
A.6.1.299 SP2.Name....315
A.6.1.300 SP2.Offset.Pos.Name....315
A.6.1.301 SP2.Offset.SP.Name....315
A.6.1.302 SP2.Offsets.Grid1.Easting....315
A.6.1.303 SP2.Offsets.Grid1.Northing....315
A.6.1.304 SP2.Offsets.Pos1.Alt....316
A.6.1.305 SP2.Offsets.Pos1.Elev....316
A.6.1.306 SP2.Offsets.Pos1.Lat....316
A.6.1.307 SP2.Offsets.Pos1.Lon....317
A.6.1.308 SP2.Offsets.WGS84.Pos1.Alt....317
A.6.1.309 SP2.Offsets.WGS84.Pos1.Lat....317
A.6.1.310 SP2.Offsets.WGS84.Pos1.Lon....317
A.6.1.311 SP2.Pos.Alt....318
A.6.1.312 SP2.Pos.Elev....318
A.6.1.313 SP2.Pos.Lat....318
A.6.1.314 SP2.Pos.Lon....318
A.6.1.315 SP2.Positioning....319
A.6.1.316 SP2.Relative.DX....319
A.6.1.317 SP2.Relative.DY....319
A.6.1.318 SP2.Relative.DZ....319

A.6.1.319 SP2.Route.Arc.DOL....320
A.6.1.320 SP2.Route.Arc.KP....320
A.6.1.321 SP2.Route.DOL....320
A.6.1.322 SP2.Route.Grid.DOL....321
A.6.1.323 SP2.Route.Grid.KP....321
A.6.1.324 SP2.Route.KP....321
A.6.1.325 SP2.Route.SeabedSlope....321
A.6.1.326 SP2.Route.Section.Bearing....322
A.6.1.327 SP2.Route.TerrainDist....322
A.6.1.328 SP2.Route.WaterDepth....323
A.6.1.329 SP2.Smoothed.CMG....323
A.6.1.330 SP2.Smoothed.Speed....323
A.6.1.331 SP2.Smoothed.SpeedKmh....323
A.6.1.332 SP2.SP1Relative.DX....324
A.6.1.333 SP2.SP1Relative.DY....324
A.6.1.334 SP2.SP1Relative.DZ....324
A.6.1.335 SP2.Speed....324
A.6.1.336 SP2.SpeedKmh....325
A.6.1.337 SP2.SpeedMS....325
A.6.1.338 SP2.Target1.Bearing....325
A.6.1.339 SP2.Target1.Range....325
A.6.1.340 SP2.WaterDepth....326
A.6.1.341 SP2.WGS84.Pos.Alt....326
A.6.1.342 SP2.WGS84.Pos.Lat....326
A.6.1.343 SP2.WGS84.Pos.Lon....326
A.6.1.344 SP3.Grid.Easting....327
A.6.1.345 SP3.Grid.Northing....327
A.6.1.346 SP3.GridHeading....327
A.6.1.347 SP3.Heading....327
A.6.1.348 SP3.LaybackBearing....328
A.6.1.349 SP3.LaybackDistance....328
A.6.1.350 SP3.LaybackMode....328
A.6.1.351 SP3.Motion.Pitch....328
A.6.1.352 SP3.Motion.Roll....329

A.6.1.353 SP3.Name.....329
A.6.1.354 SP3.Offset.Pos.Name.....329
A.6.1.355 SP3.Offset.SP.Name.....330
A.6.1.356 SP3.Offsets.Grid1.Easting.....330
A.6.1.357 SP3.Offsets.Grid1.Northing.....330
A.6.1.358 SP3.Offsets.Pos1.Alt.....330
A.6.1.359 SP3.Offsets.Pos1.Elev.....331
A.6.1.360 SP3.Offsets.Pos1.Lat.....331
A.6.1.361 SP3.Offsets.Pos1.Lon.....331
A.6.1.362 SP3.Offsets.WGS84.Pos1.Alt.....331
A.6.1.363 SP3.Offsets.WGS84.Pos1.Lat.....332
A.6.1.364 SP3.Offsets.WGS84.Pos1.Lon.....332
A.6.1.365 SP3.Pos.Alt.....332
A.6.1.366 SP3.Pos.Elev.....332
A.6.1.367 SP3.Pos.Lat.....333
A.6.1.368 SP3.Pos.Lon.....333
A.6.1.369 SP3.Positioning.....333
A.6.1.370 SP3.Relative.DX.....333
A.6.1.371 SP3.Relative.DY.....334
A.6.1.372 SP3.Relative.DZ.....334
A.6.1.373 SP3.Route.Arc.DOL.....334
A.6.1.374 SP3.Route.Arc.KP.....334
A.6.1.375 SP3.Route.DOL.....335
A.6.1.376 SP3.Route.Grid.DOL.....335
A.6.1.377 SP3.Route.Grid.KP.....335
A.6.1.378 SP3.Route.KP.....336
A.6.1.379 SP3.Route.SeabedSlope.....336
A.6.1.380 SP3.Route.Section.Bearing.....336
A.6.1.381 SP3.Route.TerrainDist.....337
A.6.1.382 SP3.Route.WaterDepth.....337
A.6.1.383 SP3.Smoothed.CMG.....338
A.6.1.384 SP3.Smoothed.Speed.....338
A.6.1.385 SP3.Smoothed.SpeedKmh.....338
A.6.1.386 SP3.SP1Relative.DX.....338

A.6.1.387 SP3.SP1Relative.DY....339
A.6.1.388 SP3.SP1Relative.DZ....339
A.6.1.389 SP3.Speed....339
A.6.1.390 SP3.SpeedKmh....339
A.6.1.391 SP3.SpeedMS....340
A.6.1.392 SP3.Target1.Bearing....340
A.6.1.393 SP3.Target1.Range....340
A.6.1.394 SP3.WaterDepth....340
A.6.1.395 SP3.WGS84.Pos.Alt....341
A.6.1.396 SP3.WGS84.Pos.Lat....341
A.6.1.397 SP3.WGS84.Pos.Lon....341
A.6.1.398 System.CommsScannerState....341
A.6.1.399 System.CoordinateSystem....342
A.6.1.400 System.Date....342
A.6.1.401 System.DBR.CablesRevision....342
A.6.1.402 System.DBR.FixfilesRevision....342
A.6.1.403 System.DBR.FixfullRevision....343
A.6.1.404 System.DBR.FixlayoutRevision....343
A.6.1.405 System.DBR.GeodeticsRevision....343
A.6.1.406 System.DBR.MobileShapesRevision. 343
A.6.1.407 System.DBR.MobilesRevision....344
A.6.1.408 System.DBR.RoutesRevision....344
A.6.1.409 System.DBR.RoutesShapesRevision 344
A.6.1.410 System.DBR.ShipRevision....344
A.6.1.411 System.DBR.VarsRevision....345
A.6.1.412 System.Time....345
A.6.1.413 System.Timestamp....345
A.6.1.414 System.VMUsage....345
A.6.1.415 Target1.Name....346
A.7 Variable attributes....347
A.7.1 heading....349
A.7.2 format....349
A.7.2.1 Numeric formats....350
A.7.2.2 Special formats....351

A.7.2.2.1 Date and time formats....351
A.7.2.2.2 Latitude and longitude formats....351
A.7.3 Variable calculation dependencies....352
A.8 Built in functions....354
A.8.1 Standard functions....354
A.8.1.1 abs(x)....354
A.8.1.2 acos(x)....354
A.8.1.3 acosh(x)....355
A.8.1.4 asin(x)....355
A.8.1.5 asinh(x)....355
A.8.1.6 atan(x)....356
A.8.1.7 atan2(y,x)....356
A.8.1.8 atanh(x)....358
A.8.1.9 bin2hex(s)....358
A.8.1.10 ceil(x)....358
A.8.1.11 chr(n)....359
A.8.1.12 cos(x)....359
A.8.1.13 cosh(x)....359
A.8.1.14 deg_offset(a, b)....360
A.8.1.15 exp(x)....360
A.8.1.16 floor(x)....361
A.8.1.17 hexdec(s)....361
A.8.1.18 iif(b,v1,v2)....362
A.8.1.19 hex2bin(s)....362
A.8.1.20 ln(x)....363
A.8.1.21 log(x)....363
A.8.1.22 ord(c)....364
A.8.1.23 sgn(x)....364
A.8.1.24 sin(x)....365
A.8.1.25 sinh(x)....365
A.8.1.26 sqrt(x)....366
A.8.1.27 strcat(s1,s2)....366
A.8.1.28 strcmp(s1,s2)....367
A.8.1.29 stricmp(s1,s2)....368

A.8.1.30 stripos(s1, s2, [index])....369
A.8.1.31 strlen(s)....369
A.8.1.32 strpos(s1, s2, [index])....370
A.8.1.33 strtolower(s)....370
A.8.1.34 strtoupper(s)....371
A.8.1.35 substr(s, start, [len])....371
A.8.1.36 tan(x)....372
A.8.1.37 tanh(x)....372
A.8.1.38 value(x)....372
A.8.2 Special functions....373
A.8.2.1 timestampOf(var)....373
A.8.2.2 historyOf(var)....373
A.8.2.3 flagsOf(var)....374
A.8.2.4 variableUpdated(var)....374
A.8.2.5 freq(hist)....376
A.9 History Objects....378
A.9.1 Properties....379
A.9.1.1 length....379
A.9.1.2 timeRange....379
A.9.1.3 rateHz....379
A.9.1.4 secondsPerSample....380
A.9.1.5 min....380
A.9.1.6 max....380
A.9.1.7 avg....380
A.9.1.8 avgmod2pi....380
A.9.1.9 avgmod360....380
A.9.2 Accessing elements....381
A.9.3 Additional functions....381
A.10 Using scripts....382
A.10.1 Defining your own constants....383
A.10.2 Defining your own functions....383
A.10.3 Updating variables conditionally....384
A.11 Reserved Words....387
A.11.1 Reserved by Javascript language....387

A.11.2 Reserved by BSPEngine....390
A.11.2.1 Vars....390
A.11.2.2 Server....390
A.12 AIS Filtering....391
A.12.1 Configuring AIS filtering....391
A.12.1.1 FilterRadius=....392
A.12.1.2 FilterUnknown=....392
A.12.1.3 FilterDeferIdents=....392
A.12.1.4 FilterInclude=....393
A.13 Some Worked Examples....394
A.13.1 Logging of vessel track and roll period....394
A.13.2 Logging of raw or filtered AIS data....397
A.13.3 Logging of system alerts....398
A.13.4 Decoding and logging more complex messages 399
B Appendix B....400
B.1 Custom panels and watch windows in Blue Spider.401
C Appendix C....403
C.1 Geodetics in Blue Spider....404
C.1.1 Extracts from various internet sources 404
C.1.1.1 WGS84 - World Geodetic System....404
C.1.1.2 EGM96 - Earth Gravitational Model 1996 406
C.1.1.3 ETRS89 - Terrestrial Reference System 1989 407
C.1.1.4 LAT – Lowest Astronomical Tide....409
C.1.1.5 TAW - Belgian national tidal reference. .410
C.1.1.6 GEBCO Gridded bathymetry data....410
C.1.1.7 MSL (Mean Sea Level)....410
C.1.2 EGM Geoid Library....411
C.1.2.1 Introduction....411
C.1.2.2 What is a Geoid....411
C.1.2.3 Geoidal Separation....414
C.1.2.4 The EGM Geoid Library Install....414

C.2 Guide to using vertical datums....416
C.2.1 Introduction....416
C.2.2 Use of KMSTrLib....417
C.2.3 GPS Receiver Datum....418
C.2.4 Datum Shifts....418
C.2.5 Vessel Offsets and Steer Points....418
C.2.6 Mobile Steer Points....419
C.2.7 Subsea depth measurement and adjustment. .419
C.2.8 Depth from pressure sensor or USBL....420
C.2.9 Ship heave sensor....420
C.3 Draft....421
C.3.1 Barometric pressure adjustment of Subsea depth readings....421
C.3.2 Barometer....423
C.3.3 Configuring the barometer input....423
C.3.4 Draft....425
C.3.5 Echo Sounder....425
C.3.6 GPS Receivers....426
C.3.7 HPR System....426
C.3.8 Subsea depth sensors....426
C.3.9 Motion sensor....426
C.4 Editing geocalc.dat to define additional datums....427
C.4.1 #Vshift=....428
C.4.2 #Vref=<Name>....428
D Appendix D....429
D.1 Bone information files....430
D.1.1 Bone editor....431
D.2 Bone scripts....435
E Appendix E....436
E.1 Vessel Simulator....437
E.2 Simulator Settings....439
E.2.1 Port Configuration....440
E.2.2 Motion Control....442

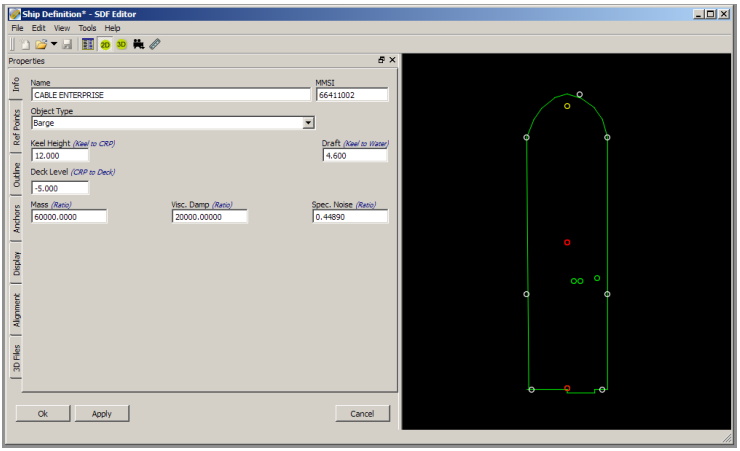
1 Introduction

This manual is intended as a User Guide. The following assumptions are made:

- That the reader has a knowledge of Windows Operating Systems
- That the Blue Spider System is already interfaced to external peripherals
- That the user is comfortable working with ini files
- That the operator is seeking to achieve the most accurate positioning possible
- That the operator wants to know how Blue Spider works

2 Configuration

2.1 Vessel Definition



Vessel, vehicle and stationary object shapes are stored in Blue Spider as .SDF (Shape Definition) files. SDF files have default names but can be saved and loaded as friendly names, for example:

1. Save File As ‘Cable Innovator.sdf’
2. Load from file ‘Cable Innovator.sdf’

SDF files can be loaded, modified or saved on any Blue Spider computer. Replication will occur to all other Blue Spider machines automatically.

Blue Spider has a built in shape definition editor but there is a replacement for this with some additional extra features which are of particular use with 3D shape definitions. If the option to install the SDF Editor is chosen in the installation options this enhanced editor replaces the one in Blue Spider and also in the Post Processing Tool if installed.

Use of the replacement SdfEdit tool is strongly recommended. The built in dialog in Blue Spider will be removed in a future release.

The full installation includes the new editor and Blue Spider and PPT will use it if installed.

SDF files contain the following information

1) Shape of the vessel or mobile (2D outline)
2) Name of the vessel or mobile
3) Offset measurements and names (POI)
4) Colours
5) Direction indicator
6) 3D shape and texture files (for 3D shapes)
7) Anchor details (Barge)
8) Proximity Alerts
9) Display options
10) MMSI number (for AIS)

By default, SDF files are known as -

In the Blue Spider folder on a Server:

a) \Blue Spider\System Config\NavShip.sdf (the ship)

b) \Blue Spider\System Config\MobileObjects (folder) the definitions for each defined mobile

c) \Blue Spider\System Config\StationaryObjects (folder) the definitions for each defined stationary object

On any machine where Blue Spider is installed

a) \Blue Spider\DataCache\LocalNavShip.sdf

b) \Blue Spider\DataCache\MobileObjects and StationaryObjects (folders) as above

In simple terms, when a ship definition file is edited or created, the SDF file is updated on the Master Server system. The file then replicates to the Slave Blue Spider system.

All instances of Blue Spider notice a change has been made and request the changed data from BSPEngine. The local copy is then updated

2.1.1 Info Tab

On the older built in editor this is labelled as the General tab. The built in editor is no longer documented and there are significant differences. Users continuing to use the built in editor do so at their own risk.

2.1.1.1 Name

Vessel Name. This needs to be unique to other vessels working in the area. The vessel name is used by the Barge Management system.

2.1.1.2 MMSI Number.

This is used to hide the vessels own AIS image.

2.1.1.3 Keel Height (Keel to CRP)

This is the measured distance from the bottom of the keel to the CRP height. The height of the CRP must have been determined prior to entering this value. The measurement is used in the calculation of sea level from GPS height.

Most of the DP systems use Keel as the vertical reference datum therefore it is likely that Blue Spider and the DP will have different antenna height offsets.

2.1.1.4 Draught (Keel to waterline)

The draught must be determined (usually from the ballasting computer) and kept up to date as the draught of the vessel changes. This measurement is used in the calculation of sea level from GPS height. Where draught is provided for bow and stern, an average figure should be put into Blue Spider.

2.1.1.5 Mass (ratio), Damping and Noise

Mass (ratio), Visc Damp (ratio) and Spec. Noise ration are used only for the Kalman filtering option. This is seldom used so the values can be left as defaults.

2.1.1.6 Minimum display size

This is used to ensure that the vessel shape is always visible on the Blue Spider screen. It refers to the size of the vessel in cm as seen on the plan view. If the minimum size is used, then the vessel will start to flash, indicating that the vessel shape is no longer to scale.

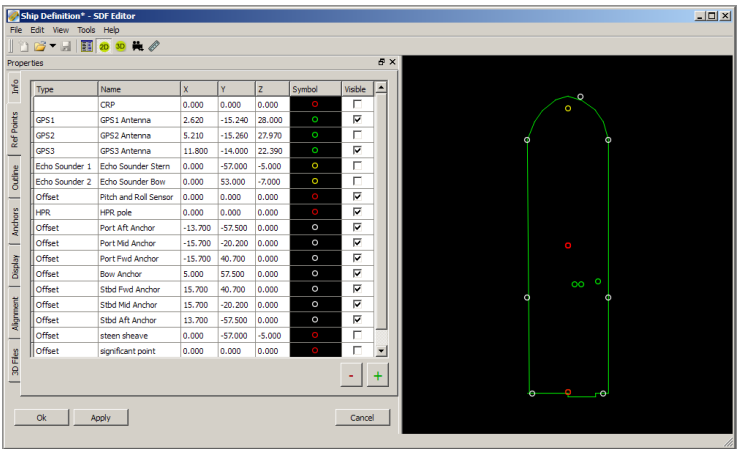
This feature has no effect on the 3D Viewer.

2.1.1.7 Save Definition and Load Definition

The file menu allows definitions to be saved and loaded.

2.1.2 Ref Points Tab

Points of interest are used to determine precise locations of navigational equipment or points on the vessel to track. The positional accuracy can be entered to a resolution of millimetres.


Ship Definition* - SDF Editor
File Edit View Tools Help
Properties
Info
Ref Points
Outline
Anchors
Display
Alignment
3D Files
Type Name X Y Z Symbol Visible
CRP 0.000 0.000 0.000
GPS1 GPS1 Antenna 2.620 -15.240 28.000
GPS2 GPS2 Antenna 5.210 -15.260 27.970
GPS3 GPS3 Antenna 11.800 -14.000 22.390
Echo Sounder 1 Echo Sounder Stern 0.000 -57.000 -5.000
Echo Sounder 2 Echo Sounder Bow 0.000 53.000 -7.000
Offset Pitch and Roll Sensor 0.000 0.000 0.000
HPR HPR pole 0.000 0.000 0.000
Offset Port Aft Anchor -13.700 -57.500 0.000
Offset Port Mid Anchor -15.700 -20.200 0.000
Offset Port Fwd Anchor -15.700 40.700 0.000
Offset Bow Anchor 5.000 57.500 0.000
Offset Stbd Fwd Anchor 15.700 40.700 0.000
Offset Stbd Mid Anchor 15.700 -20.200 0.000
Offset Stbd Aft Anchor 13.700 -57.500 0.000
Offset steen sheave 0.000 -57.000 -5.000
Offset significant point 0.000 0.000 0.000
Ok
Apply
Cancel

It is essential that equipment used to measure positional information is defined by the 'Type' in this table, as well as entering the X, Y and Z information for each offset.

An exception to this rule applies if the HPR offsets are applied in the HPR. (Usually the case). When the HPR pole offsets are applied in the HPR, **the 'Type' in Blue Spider must be set to 'Offset'** or the offsets in Blue Spider set to zero or else the offsets would be applied twice.

The first option is favourable since it is convenient to see where the HPR pole offset is on the vessel.

2.1.2.1 Types of Offsets

- **Offset** Used for any SP being tracked
- **GPS1** Defines GPS system 1
- **GPS2** Defines GPS System 2
- **GPS2** Defines GPS System 3
- **HPR** ONLY when offsets are not applied by the HPR
- **Echo Sounder** Defines Echo Sounder in use (for ship definitions only)
- **Depth Sensor** Defines a pressure sensor offset (mobile objects only)
- **Beacon** Defines a beacon offset offset (mobile objects only)

2.1.2.2 Pitch and Roll

The pitch and roll is not included on this table. The position of the motion sensor is generally close to the vessels CRP and close to the waterline.

Pitch and roll information is used in conjunction with the GPS and gyro data to calculate lever arm movements. Correct offsets and more importantly, sense (Invert Pitch / Roll) are critical when working to high levels of accuracy (cm).

Pitch and Roll corrections are entered into the system in the Naverv.ini file in the [RP] Section.

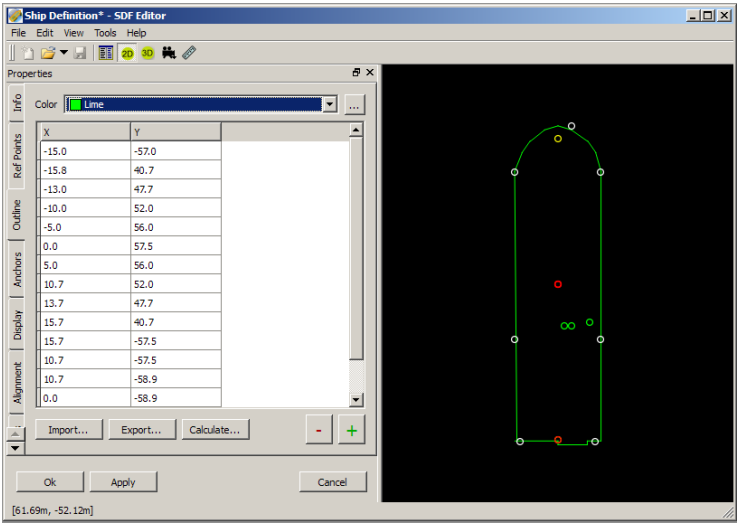
```
[RP01]
InvertRoll      = 0
InvertPitch     = 0
PitchOffset     = 0.15 ; Calibration 30 Jan 2012
RollOffset      = 0.63 ; Calibration 30 Jan 2012

[RP02]
InvertRoll      = 0
InvertPitch     = 0
PitchOffset     = 0.00
RollOffset      = 0.00
```

See A.3.1.9 [RP01]...[RP03] (p.182)

2.1.3 Outline Tab

The vessel outline made up of points around the ship which form the outline. All vessels, mobiles, and stationary objects should have an outline even if it is only a crude approximation.



The vessel shape is created on a grid, where +X is right and +Y is up. Points are created sequentially around the outside of the vessel to create the ship shape. The X and Y readout (bottom left) help create the shape, but it is easier to plan the vessel shape coordinates on paper initially.

The outline colour can also be set here.

You can import an outline from file or if you already have a 3D definition an outline can be created automatically using the Calculate button. This builds an outline by tracing the outline of the 3D shape. However before considering the use of this option you need to ensure that your 3D model is correctly aligned. If your 3D model drawing origin is identical to the CRP then this is not a problem but more often than not the origin of the model might not coincide exactly. Aligning a 3d model is easiest with an existing outline and POI points.

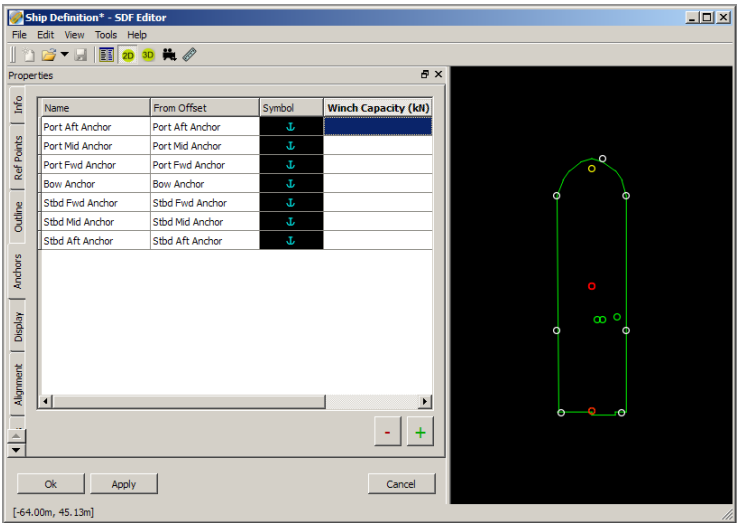
For more information see 2.1.6 Alignment Tab (p.44)

2.1.3.1 Saving and loading an outline

The outline can be imported and exported as a simple CSV file containing just the X and Y offset values. This is useful if you want to use 3rd party tools to edit the outline. The outline is a closed polygon and the units are in metres with 1cm resolution.

2.1.4 Anchors Tab

This page is used to determine anchor winch locations on this vessel.



The anchors are created here but they are actually positioned by 'Offsets' from the 'Points of Interest' table. Therefore there must be new point of interest created for each anchor point.

There are 3 Right Click options

- Append
- Insert
- Delete

It is recommended to use the same name for the offset as for the anchor. This helps avoid confusion. Anchors can be represented by coloured symbols (e.g. red and green). The anchor symbol refers to the symbol that will represent the anchor once it is deployed.

The anchor locations should be given a meaningful name; this might be a numbered sequence or names like 'Port-Fwd', Stbd-Fwd'.

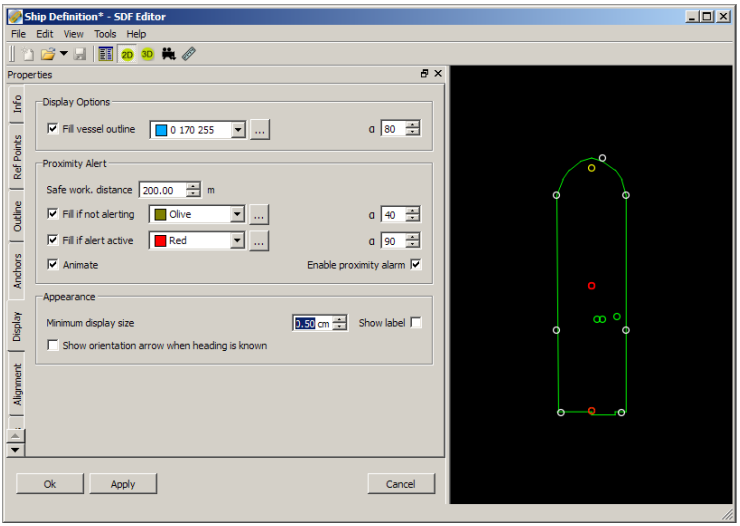
It is worth keeping anchor names reasonably short as this keeps display of anchor related data more compact and easier to read.

In addition if you have interfaced to equipment to give you support for anchor winch payout length, tension and speed. You can also specify the holding capacity (maximum tension) for each winch. If you are not using this you can leave blank.

For more information see 4.5 Anchor Winch Support (p.142)

2.1.5 Display Tab

This page determines how the vessel is seen on this screen and on other screens, including those linked by the Barge Management system.



2.1.5.1 Display Options

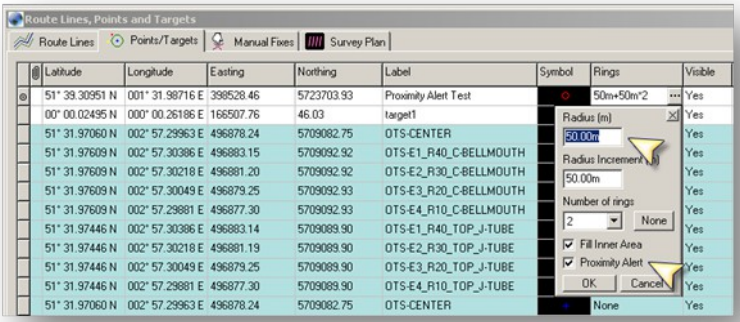
The vessel outline can be filled with a solid or transparent colour. The Alpha figure determines the transparency of the colour fill. This feature only applies to the 2D vessel shape. For use with the 3D viewer a 3D model would normally be present.

2.1.5.2 Proximity Alert

This feature is very useful when vessels equipped with Blue Spider are working in close proximity to each other or when vessels have to work close to stationary objects. The system works by establishing a safe working distance from this vessel to another vessel or fixed point. If the proximity area is breached, a red circle will be shown around the vessel that has the alarm enabled.

It is recommended to use the 'Fill when alert is active' feature to ensure the alert is not missed. The colour and transparency of the warning circles can be changed. There is also an option to animate (flash) the warning circle.

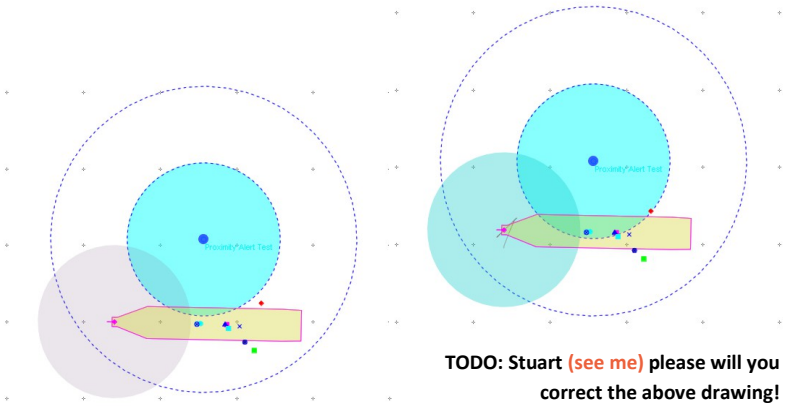
Creating a Proximity warning around a stationary point (Target) is carried out in the **Points and Targets** table. The arrows indicate the size of the circle and the options available.



In this screen shot, 2 range rings represent 50m and 100m around the target. The vessel has a proximity alarm of 50m. The alarm on the left diagram is not yet activated since the CRP of the vessel is not yet within the 50m proximity zone of the target. If the CRP of the vessel passes within 50m of the target, then the alarm is activated, as shown on the right diagram.

The alert is indicated by the change of colour of the ring around the vessel offset.

NOTE that the alert is triggered from the steerpoint of the vessel to the target, not the closest point of approach. If the target is another vessel then the circles are around each vessels current steerpoint. This may be modified in a later version of Blue Spider.



TODO: Stuart (see me) please will you correct the above drawing!

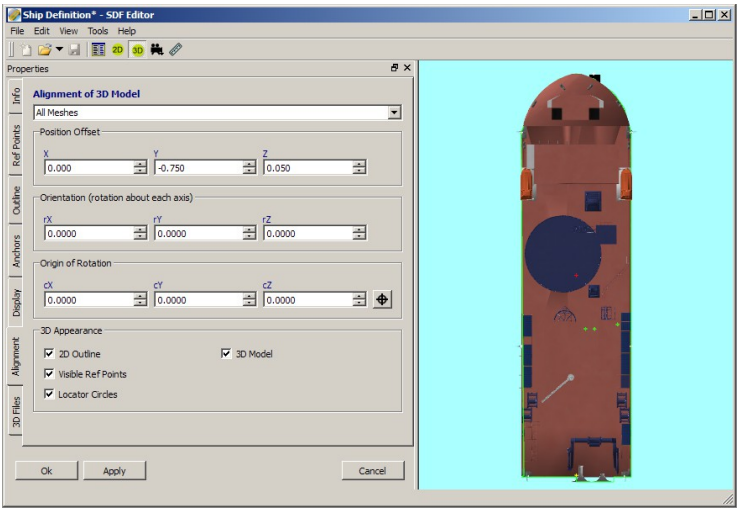
2.1.5.3 Appearance (Minimum Display Size)

This option only pertains to the 2D display in Blue Spider. If selected then if you zoom out on the plan view such that the ship would disappear to a tiny speck then instead the ship is drawn at the specified size (in cm) but is also animated so it grows from a tiny point to the minimum size. This feature makes it possible to quickly locate the ship when zoomed right out.

This feature does not apply to the 3D viewer where instead you can use the locator circles option as an alternative.

2.1.6 Alignment Tab

The alignment tab allows a 3D model to be placed in the correct position with respect to the 2D outline and offsets.



If you have more than one mesh file comprising your model you can individually align them (by selecting the mesh name) or align them all together (by selecting **All meshes**).

2.1.6.1 Position Offset

The position offset is the offset from the origin of the 3D model to the CRP of the vessel. By turning on 3D display mode (after adding the necessary 3D files) you can visually align the 3D model with the 2D outline and reference points. It can be difficult to align models correctly particularly if a rotation is needed as well. Practice is needed in order to be able to do this correctly and quickly.

2.1.6.2 Orientation

Orientation specifies the rotation amount about each axis. In most cases models will already be aligned to the usual vessel reference frame. If you need to enter rotation values then you are likely to also need to work out the rotation origin.

2.1.6.3 Origin of rotation

The origin of rotation specifies the origin that rotation should be applied to. By default this is the (0,0,0) location in 3D model coordinate space. By pressing the little cross-hair button you can shift this to the central location. This may help for models where the actual drawing location is quite distant from the actual vessel CRP. Some models however may need rotation applied to an entirely different location. Models like this may take considerable practice to align correctly.

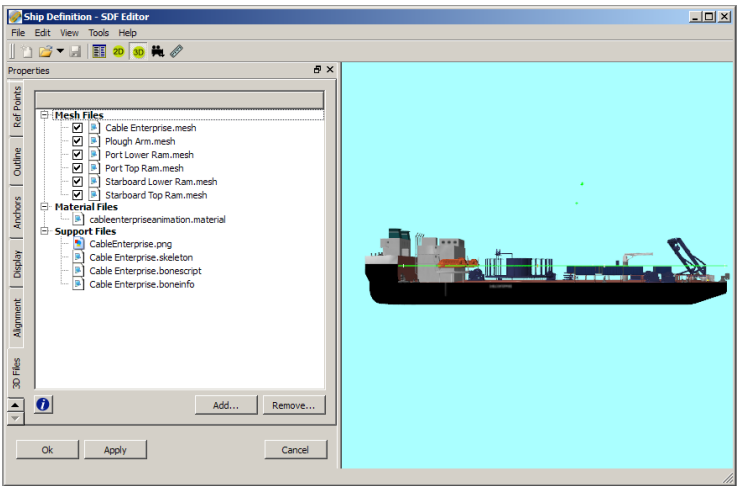
2.1.6.4 3D Appearance

This section controls the appearance in the 3D viewer

- **2D Outline**
  This specifies whether or not to draw the 2D outline.
- **Visible Ref Points**
  Specifies whether or not the reference points should be displayed. Only ref points marked as visible will actually be displayed.
- **Locator Circles**
  If this option is selected then concentric circles at increasing radii are drawn around the vessel making it easier to locate from a distance.
- **3D Model**
  If this option is not selected then the 3D model will not be displayed. This option will be greyed out if there isn't a 3D model in the definition.

2.1.7 3D Files Tab

The 3D files tab lets you add a 3D model to the definition



2.1.7.1 Mesh files

A model is made up of one or more mesh files. In the screenshot above there are several separate mesh files but they are are used as if there is just one. There are options on this page (the check boxes next to each mesh file) to allow individual meshes to be hidden. This option is only provided here so you can see which parts of the model belong to the different mesh files. In the 3D viewer all meshes will be displayed regardless of the state of these check boxes.

2.1.7.2 Material files

A model will typically have just one material file. The material file defines the appearance of the surface area of the ship. Typically it will also reference one or more image files to provide a texture-mapped surface. Image files have to be referenced in the material file and have to be added as support files.

To use texture mapping the mesh file must also contain texture mapping coordinates for every vertex.

Alternatively the mesh can be arranged with groups of vertices each being assigned a single material name in the material file which means different areas can be given different colours.

It depends on how the mesh was produced as to whether will be able to use texture mapping or not.

2.1.7.3 Support files

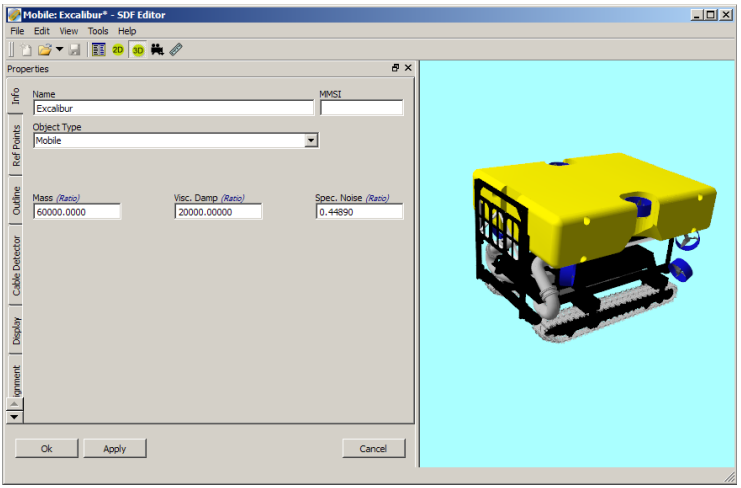
Support files include any images (typically .png images) that are used and referenced by the material file.

In addition to images for texture mapping, support files can include the following:

- **Skeleton**
  A skeleton connects moving parts of a model together. A skeleton can link separate groups of vertices and define where they join. Mesh files have to be prepared with bone assignments to use a skeleton.
- **Boneinfo**
  A bone information file provides additional information to Blue Spider. It can be created from the skeleton using the SDF editor. You need a boneinfo file in order to animate moving parts of a model. You can't have a boneinfo file without a skeleton.
- **Bonescript**
  A bone script is only needed for complex animation. For a simple hinge or extending part you are unlikely to need a script but if many parts move in a complex way then a script can compute the bone vertex positions based on a simple input variable (or more than one)

Information on bone animation and scripts can be found in Appendix D (p.429)

2.2 Mobile Definitions



Editing mobiles is similar to editing vessel definitions and there are only a few subtle differences in the options available

Vessel (and mobile) shapes are stored in Blue Spider as .SDF files. SDF files have default names but can be saved and loaded as friendly names, for example:

1. Save File As ‘ST200.sdf’
2. Load from file ‘ST200.sdf’

SDF files can be loaded, modified or saved on any Blue Spider computer. Replication will occur to all other Blue Spider screens automatically.

2.2.1 Info Tab

2.2.1.1 Name

This is the name given to the Mobile. If you change the name of the mobile you will also rename the mobile as it appears in the mobiles database.

2.2.1.2 MMSI

The MMSI code does normally not apply to mobiles. If however you do have AIS tracking capability you can enter the MMSI number in order to prevent the AIS object from being displayed.

2.2.1.3 Object Type

This is always 'Mobile' when configuring a mobile sdf. There are no other options.

2.2.1.4 Mass

Mass, Visc. Damp and Spec. Noise can all be ignored for mobiles.

2.2.1.5 Appearance

Use the **Minimum display size** to ensure that the mobile will always be visible. This feature is useful when the plan screen is zoomed out.

Show orientation arrow is a useful feature to enable when working with mobiles. If the arrow is not visible on the ROV there are two likely reasons;

1) The ini file does not map the heading input to ROV1.Heading or ROV2.Heading.

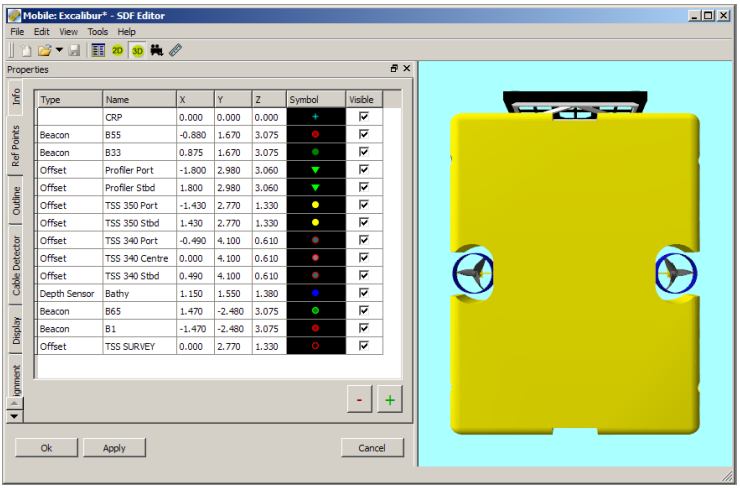
2) The heading information is not being received.

2.2.2 Outline Tab

The outline tab for mobiles is the same as the outline tab for a ship. Refer to 2.1.3 Outline Tab (p.36)

2.2.3 Points of Interest Tab

Points of Interest are defined for the mobiles. These include the beacon positions.


Mobile: Excalibur* - SDF Editor
File Edit View Tools Help
Properties
Info
Ref Points
Outline
Cable Detector
Display
Type Name X Y Z Symbol Visible
CRP 0.000 0.000 0.000
Beacon B55 -0.880 1.670 3.075
Beacon B33 0.875 1.670 3.075
Offset Profiler Port -1.800 2.980 3.060
Offset Profiler Stbd 1.800 2.980 3.060
Offset TSS 350 Port -1.430 2.770 1.330
Offset TSS 350 Stbd 1.430 2.770 1.330
Offset TSS 340 Port -0.490 4.100 0.610
Offset TSS 340 Centre 0.000 4.100 0.610
Offset TSS 340 Stbd 0.490 4.100 0.610
Depth Sensor Bathy 1.150 1.550 1.380
Beacon B65 1.470 -2.480 3.075
Beacon B1 -1.470 -2.480 3.075
Offset TSS SURVEY 0.000 2.770 1.330
Ok
Apply
Cancel

For positioning, there are two offsets to look out for on mobiles.

1) The SteerPoint Offset (the point we are logging)
2) The Positioning Offset (the position of the beacon(s))

If these are not correct, errors very quickly show up when the ROV is on a different heading. In practice this can mean that tracking up a cable one way might give a 4m positional difference to when tracking the cable in the opposite direction.

2.2.3.1 Beacon Names and Offsets

Blue Spider will attempt to match the Beacon ID with the correct offset when beacons are selected. It does this by looking for the Beacon ID in the 'Description' of the Point of Interest. Therefore, name the 'Descriptions' the same as the Beacon codes. This works well as long as the beacons are not inadvertently moved from their offset location.

This feature is often overlooked and can cause ROV / Plough offset errors if not set up correctly.

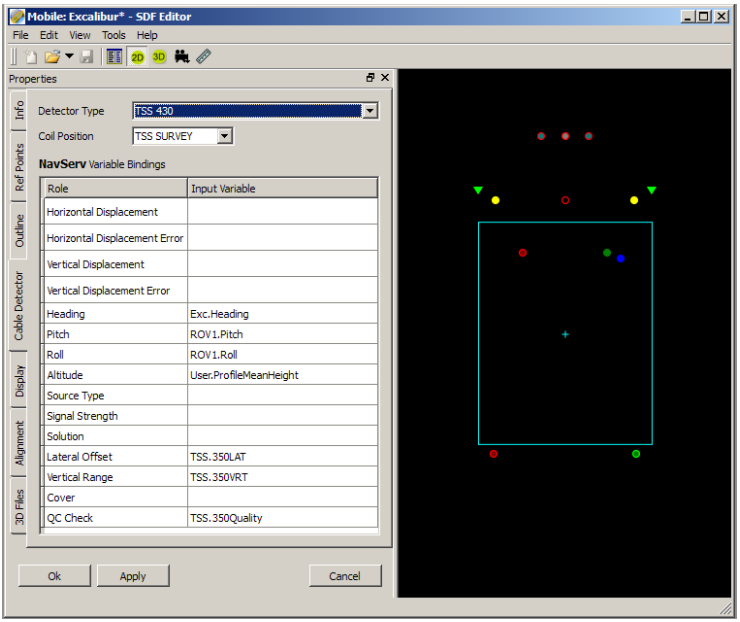
This feature does not remove the responsibility from the surveyor to check that the correct offsets are selected.

2.2.3.2 Depth Sensor Offset

Selection of the Depth Sensor offset automatically overrides depth information from the HPR beacons. If there is no Depth Sensor offset, then Blue Spider will compute the depth of the vehicle from the HPR z value and the Beacon z offsets. But if it does this it will raise an alarm as well.

2.2.4 Cable Detector Tab

For mobiles an extra tab is available to allow configuration of a cable detector (if one has been fitted)



If you don't have a cable detector fitted then just set the type to None.

If you do have a cable detector and its either a TSS340 or 350 (or both) then you may need to configure it here.

For a TSS the Lateral Offset and Vertical Range variables should be set to the input variables that bring in these values from the input message from the unit.

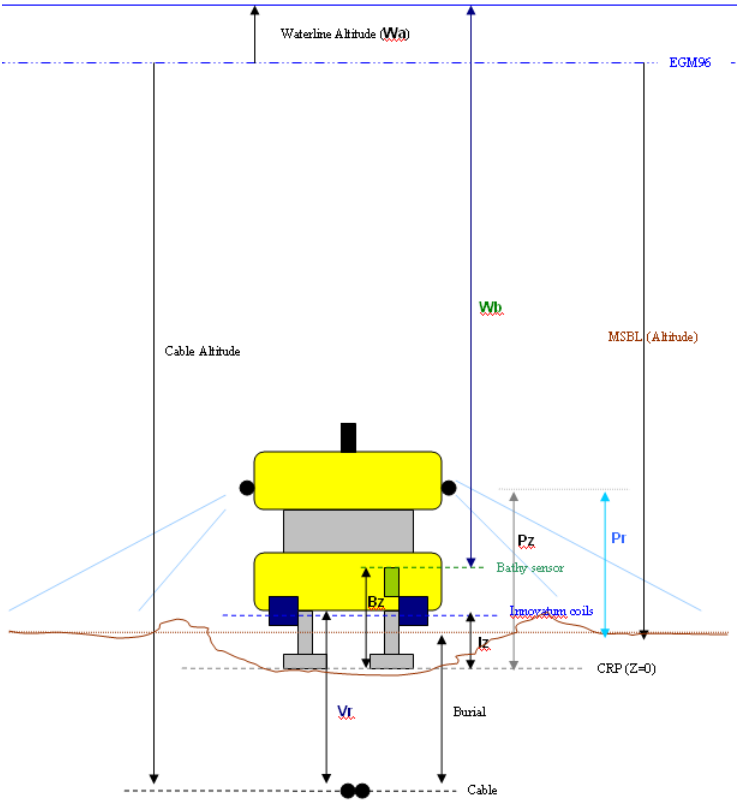
For Innovatum Smarttrak enter variable names for the Horizontal and Vertical displacements.

The other variables listed in the table can be specified but they are not actually used.

The coil position needs to be specified as an offset. So make sure you have added one to the Ref Points.

2.2.4.1 How burial is calculated

The burial calculation fairly straightforward but it does take the attitude (pitch and roll) of both the mobile and the ship into account. The following diagram illustrates essentially how the calculations are performed.



Where:

Wb Raw water depth from bathy sensor on ROV

Pz Profiler Z offset from CRP

Iz Innovatum Z offset from CRP

Bz Bathymetry Z offset from CRP

Vr Vertical range to cable from Innovatum coils

Pr Profiler range from heads to mean seabed

Wa Water line altitude from vertical datum e.g. EGM96

Blue Spider computes MSBL altitude as follows:

```
MSBL  altitude  = Wa - (Wb + Bz - Pz + Pr)
```

Blue Spider computes Cable altitude as follows:

```
Cable altitude  = Wa - (Wb + Bz - Iz + Vr)
```

Burial depth is simply:

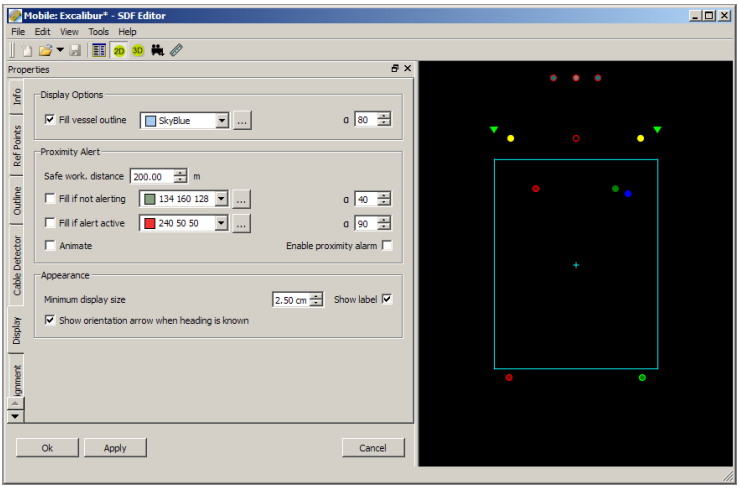
```
Burial          =  (Vr - Iz) + (Pz - Pr)
```

The waterline altitude Wa is computed by taking the altitude reading from the GPS receiver(s), then adjusting this by compensation for pitch & roll and the antenna X,Y,Z offsets to the ships CRP (central reference point). Note that the effective (altitude difference) distance between the CRP (plane) and the antenna will shorten or lengthen as the ship pitches and rolls. If there is zero pitch and roll this adjustment is simply the height of the antenna. An

adjustment is then made to this computed CRP altitude to bring this level down to the actual waterline e.g. compensation for draft and heave.

2.2.5 Display Tab

The display tab is used to define how the mobile is seen and for setting up the Proximity Alerts. These are configured in the same way that vessels are defined, and is explained in the Vessel Definition section of this manual.



All of the options that apply to vessel definitions are also available for mobiles.

See 2.1.5 Display Tab (p.40)

2.2.6 Alignment Tab

This is identical to that for a vessel.

See 2.1.6 Alignment Tab (p.44)

2.2.7 3D Files Tab

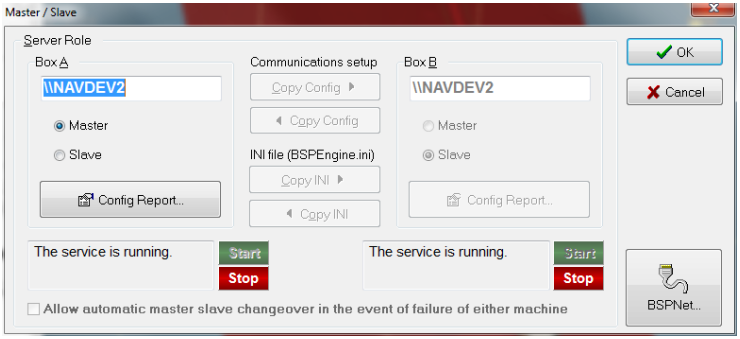
This is identical to that for a vessel.

See 2.1.7 3D Files Tab (p.47)

Note: Bone animation can also be used for mobiles and even for stationary objects. It works in the same way for all and just requires connection of BSPEngine variables to bone information (optionally via a script)

2.3 Master / Slave

The Master / Slave dialog box provides clues that the primary system and the backup system are both healthy. The dialog box should look like this:-



'The RPC server is unavailable' means that the server is either switched off, is not connected to the network, cannot be seen by this computer or for some other reason, cannot be seen by this computer, for example, Active Directory or User Account problems. This is not an acceptable condition if the Master / Slave redundancy is required and must not be ignored in this case.

The Master / Slave automatic changeover option will only be available when both servers are available.

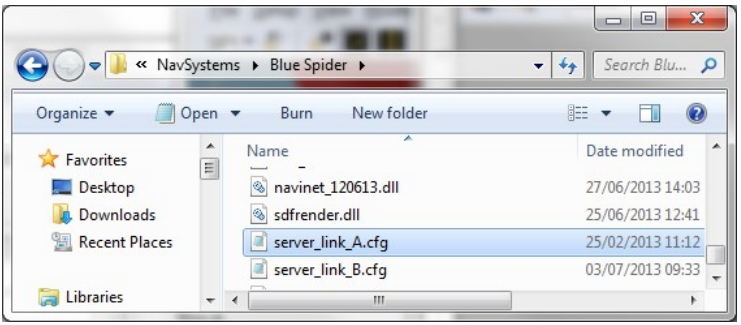
In the case of tug installations, or other vessels running a Master server only, this **'The RPC server is unavailable'** will be seen on the BSPEngine B side of the dialog box.

2.3.2 Master Slave Switch

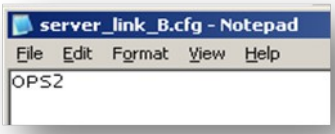
The Master and Slave are selectable from the radio buttons, and whilst it is possible to select two Masters or two Slaves, the Blue Spider system will alert all the computers on the network if one of these undesirable options is selected, with a banner message.

2.3.3 Box A and Box B

These represent the servers that 'this' Blue Spider is referring to for its Master and Slave. The selections of the correct server(s) are not user changeable in the dialog box but are set by text files in the Blue Spider folder.



With the introduction of Barge Management and stand alone servers as well as OPS1 and OPS2 on the same vessel, it is imperative to have these files configured correctly.



The files are not required in a basic 2 server (master/slave) arrangement, Blue Spider will attempt to default to OPS1 and OPS2 if they are not there, however it is good practice to always add these files.

Note. These files DO NOT go into the System Config folder. This is because, if Blue Spider is installed as Blue Spider client

only, (with no BSPEngine) then the System Config folder will not exist.

Note that although the term server is used for Box A and Box B these machines do not have to actually run a windows server operating system and can be normal workstations. BSPEngine can run on a normal workstation or on a windows server machine.

2.3.4 DNS

Blue Spider uses DNS (Domain Name Service) to resolve computer names to IP addresses. For example

OPS1 (computer name) **10.10.57.1 (IP Address)**

The DNS Server (A Windows Service) runs on both Servers (OPS1 and OPS2) and they are, and should remain synchronised. The resolution of a computer name may be served by either Server.

Problems can exist if the DNS becomes corrupted or the two DNS servers lose synchronisation with each other. The effects may be that computers can be pinged by IP address but not by name. Another effect is that Box A may not be able see Box B and visa versa.

In particular server machines that have been in storage for long periods of time (not switched on for a few months) are likely to suffer from DNS issues and may require maintenance.

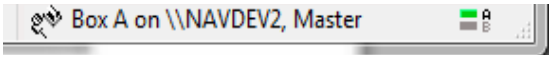
If it appears that computers cannot be contacted by name, including in VNC, then it is likely that DNS problems exist.

DNS problems should be taken seriously and addressed immediately. DNS is not part of Blue Spider software.

2.3.5 Single Installation Box A (One Server)

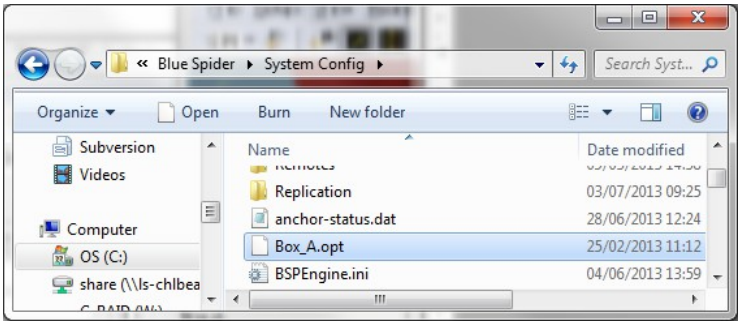
Two important configuration files are

- box_A.opt and
- box_B.opt



These files contain nothing. One of them simply has to exist if BSPEngine needs to be forced into an A or B state.

The status (A or B) is determined either in the server dialog box or at the bottom of the Blue Spider screen. In this example, box A is Master and box B is not available (a typical tug configuration). It is rare to force a single computer into the B state.



Because this file is used by BSPEngine, it must be placed in the System Config folder.

2.3.5.1 Copy Config

Copy Config will transfer all the communication port settings from one server to the other. Ensure that the correct direction is about to be used. If Box A is Slave and Box B is Master, then the top button will transfer all the OPS2 settings to OPS1. If you have fully configured OPS1, and were intending to transfer all the settings to OPS2, then you will be most displeased if this was the wrong way around.

OPS1 is the computer name and OPS1 can be Master or Slave.
OPS2 is the computer name and OPS2 can be Master or Slave.

Settings are only generally transferred if the communications ports are on another computer. (not OPS1 or OPS2). This is because OPS1 and OPS2 need to both be able to work in the event of a total power failure on a server. For this reason, the ports are generally configured as below.

Device	OPS1	OPS2	
GPS1	OPS1\COM3	OPS2\COM3	
GPS2	OPS1\COM4	OPS2\COM4	
GPS3	OPS1\COM5	OPS2\COM5	
Cable Engine	PortEng\COM7	PortEng\COM7	

Because the cable engine is on a remote PC (not OPS1 or OPS2) it is common to both servers.

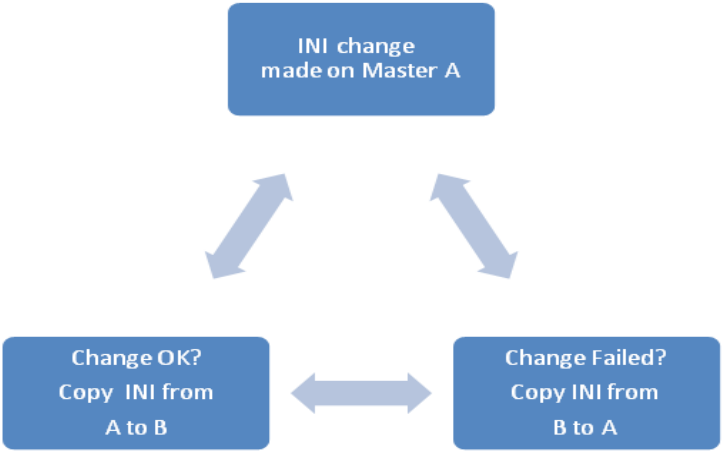
2.3.5.2 Copy INI

When changes are made and saved to the BSPEngine.ini file, BSPEngine detects that the file on Box A is now different to the file on Box B.

An alarm is raised showing that there is a difference between the two files.

The Copy INI button copies the BSPEngine.ini files either from A to B or B to A. Therefore if changes have been made on Box A and they did work, click on A>B.

In the unlikely event however the changes made to the BSPEngine.ini file on Box A have caused problems, click B<A to restore the original BSPEngine.ini file.



2.3.5.3 Manually Copied Config Files

Not all the required configuration files are automatically copied from Master to Slave.

If these files are modified, they must be manually copied from Master to Slave.

- Machine.acl.ini
- NavFix.cfg
- PortManifest.cfg

2.3.5.4 Machine.acl.ini

This file contains the address and permissions of computers that are allowed to

- See latitude and longitude positions on the screen.
- Modify geodetic settings.
- Record anchor handling operations.

This is documented in detail in Appendix A, section A.4 (Machine.acl.INI) (p.206)

2.3.5.5 Navfix.ini

This file is used to configure the contents of the Fix Dialog box. IT is modified through the file called Navfix.~cfg and is converted to binary when saved.

2.3.5.6 Nav-Inet.ini

This file is used to configure the Blue Spider Web service to the outside world.

2.3.5.7 PortManifest.cfg.

This file is used to modify the titles and the existence of the the COM port tabs for serial inputs. It is modified through the PortManifest.~cfg file and is converted to binary when saved.

Failure to ensure that these files are the same as on the Master Server could lead to problems if Box B has to become Master.

2.3.6 Automatic Master / Slave

If a BSPEngine server fails, the automatic server changeover can be used to ensure uninterrupted service by taking over (becoming Master) and taking data from the other set of COM ports on the other computer.

If Copy Config is used without due care, then all the settings on OPS2 may refer to COM ports on OPS1. This would be a problem if OPS1 failed.

2.4 Data Communications

2.4.1 BSPNet

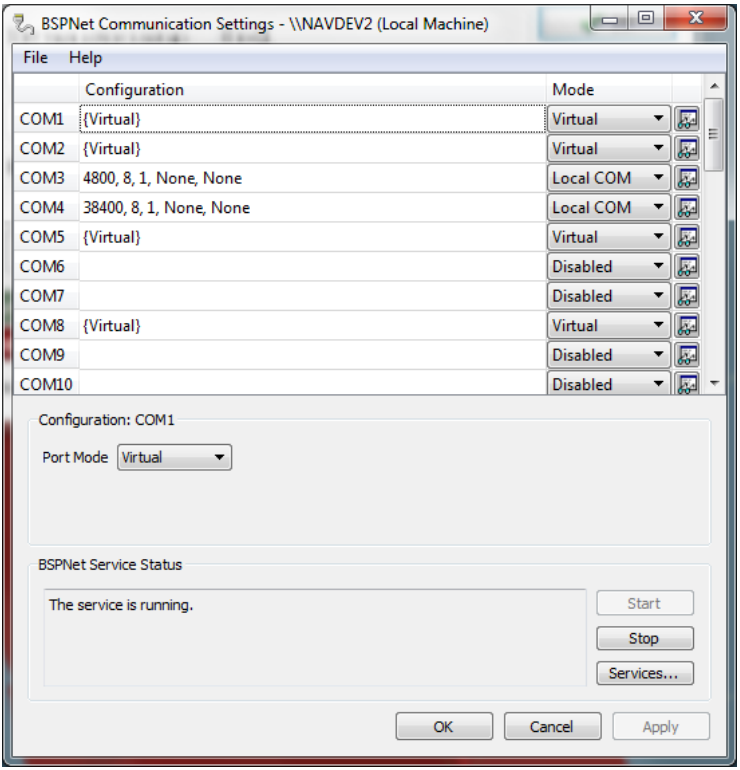
All serial data into and out of BSPEngine is routed through BSPNet. BSPNet is a standalone application that runs on any computer that uses it's COM ports to transfer data.

- BSPNet is a 'Service'. It requires a User Account and a Password to start.
- By default, BSPNet starts up automatically when the PC is booted but it can be manually started through the Windows 'Services' or from the BSPNet page in Blue Spider.
- A software upgrade forces BSPNet (and BSPEngine) to stop. The services must be manually restarted, or the PC restarted after an upgrade.
- BSPNet locks enabled ports of the PC. If BSPNet owns a COM port, then other applications, like Hyper Terminal, won't be able to use that same port.
- To free up a port for Hyper Terminal to use, it is only necessary to disable that one same port in BSPNet. There is no need to stop the BSPNet service completely.
- BSPNet can create Virtual Com Ports for use with the Data Simulator.
- The port monitor can be used to view raw data.

BSPNet and BSPEngine can be started from Services.

Only enable the COM ports that BSPNet requires on each PC. The other ports will then be available for other applications.

BSPNet ports are bi-directional.



2.4.2 Configuring Ports

COM ports are assigned to Channels in Blue Spider.

Survey input channels are predefined and other Channels are user customisable. The predefined channels are on the left and the custom channels are on the right of the tabs.

Even though a channel (GPS1) for example, is a predefined (System) channel, it is possible to alter its name. This is done using two special files in the System Config Folder.

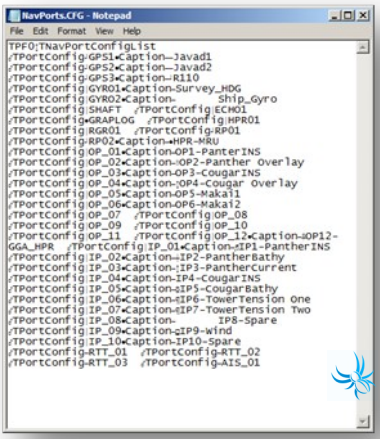
2.4.3 Editing Channel Names

All the COMS Channel tabs have default names, i.e

NavPorts.CFG

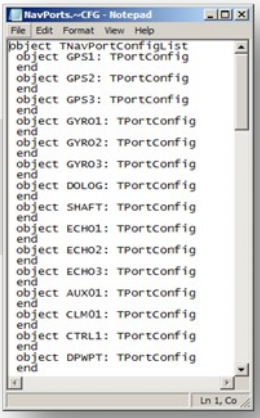
NavPorts.~CFG

"GPS 1" but these names can be changed to suit.



Two files are used to edit Channel names. They are found in the Blue Spider\System Config folder and are called **PortManifest.cfg** and **PortManifest.~cfg**.

PortManifest.cfg is a binary file and is not easily editable. However, we also have text file, PortManifest.~cfg which is editable as shown here.




```
object GPS2:TPortConfig
   Caption =  'Javad-2'
end
```

The .~cfg file can edited as required (in Notepad) and then has to be saved saved as PortNavPorts.cfg. After you have saved the file you need to restart BSPEngine and on startup it will convert the file to a binary format if it has been edited.

The original text version will be automatically copied to PortManifest.~CFG.

The PortManifest is one configuration that cannot be changed while the server is running.

If a caption is not specified for a port a default built in caption will be used.

If the editing in Notepad was not carried out correctly, then, when saving the files, it will be ignored and will not actually change! If this happens, return to the PortManifest ~cfg and look for errors.

An alarm will be raised if there are errors loading this file.

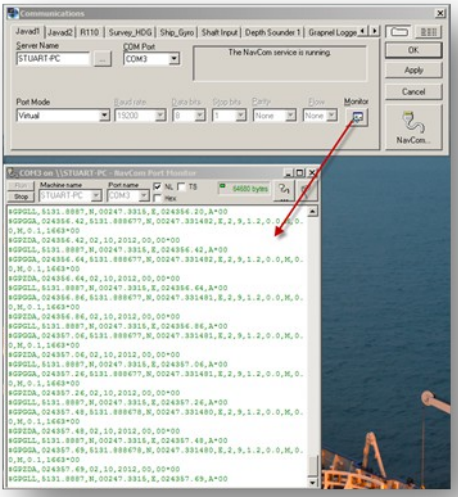
If this file does not exist then a default one will be created when the BSPEngine starts up. If you have made a serious mistake in editing this file and do not have a back-up then deleting these files is one way to recover but you will loose any changes you have made.

2.5 Monitoring Ports

Click on the Monitor button in the Communications page to open the Port Monitor. The data is observed here as BSPEngine sees it. If the serial settings are incorrect (wrong baud rate for example) then the data will appear garbled, and it will not be decoded by BSPEngine.

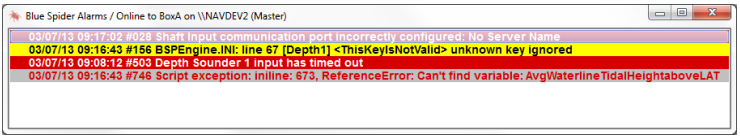
Binary data may appear to be garbled data when viewed here, possibly leading to some confusion. It is important to know if the source of the data is in binary. For example, the HPR410 message can be in binary as can some motion devices.

Most serial messages are in an ASCII NMEA format and the data is easily readable in the Port Monitor.



2.6 Decoding Alarms Message

Alarms are there to help, and there is little point in having the alarm panel minimised or not open at all. If data stops coming in to Blue Spider, then data won't be logged.


Blue Spider Alarms / Online to BoxA on \\NAVDEV2 (Master)
03/07/13 09:17:02 #028 Shaft input communication port incorrectly configured: No Server Name
03/07/13 09:16:43 #156 BSPEngine.INI: line 67 [Depth1] <ThisKeyIsNotValid> unknown key ignored
03/07/13 09:08:12 #503 Depth Sounder 1 input has timed out
03/07/13 09:16:43 #746 Script exception: iniline: 673, ReferenceError: Can't find variable: AvgWaterlineTidalHeightaboveLAT

Alarms are displayed with the Date and Time, Code and followed by the location and nature of the problem.

In the above example we have several alarms.

The first is warning us that we have a shaft input (cable channel) defined in the port manifest but we have not actually configured the port to e used in the communication settings. Either we should remove this port from the manifest (if were not using it and this will require a restart of the service) or we should just configure the communication settings for it.

The second is a warning regarding the configuration BSPEngine.INI. In this particular case a line has been added to the file that is not recognized.

```
[Depth1]
MsgName                 = $SDDBT
MsgType                 = 1,0,6
WaterDepthMetres        = 4,0,0
ThisKeyIsNotValid = 666
```

The 3rd warning tells us that we are no longer recieving data from the echo sounder. Maybe a cable has become unplugged so we need to check.

The 4th Warning tells us that we have a variable referenced in an expression in the INI file that does not actually exist. We should check the INI file and correct this by either adding the missing variable or by correcting a misspelt name.

2.6.1 Decoding Alarm Colours

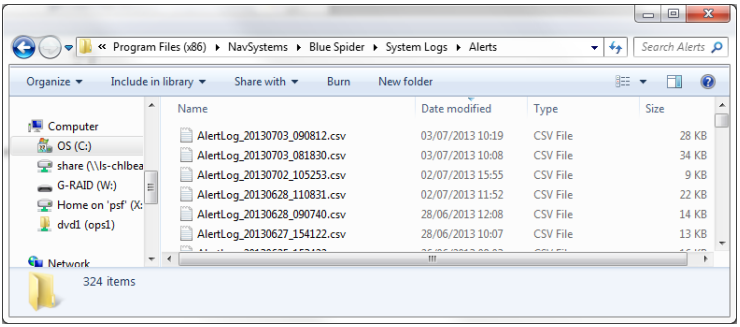
Alarms are shown in three colours.

Colour	*Meaning*
Grey	The alarm was raised but the reason the alarm appeared has now gone. Grey alarms are very common after system start up while the system establishes communications with COM ports. They will also appear if remote computers running BSPNet are shutdown or restarted.
Yellow	Yellow alarms are 'Warnings'. They are important as they need to be dealt with, but they are not stopping the system from working. They will appear if the geodetics show that the vessel is in the wrong UTM zone, or if there are sections in the ini file that are incorrect, duplicated or mis-typed. Yellow alarms will clear if the Alarms are cleared, but yellow alarms relating to errors in the ini file WILL NOT REOCCUR until BSPEngine is restarted again. Clearing these alarms DOES NOT fix the problems they are reporting.
Red	Critical errors ranging from loss of data, no access to logging files, too much memory usage or lack of hard disk space. Red Alarms MUST be fixed.

All alarms (and other entries) are logged in the Alert Logs.

2.7 Alert Logs

All alarms that appear in the alarm panel are also written to the Alert Logs (shown below). The Alert logs are an extremely useful source of information for determining when a problem started to occur. They can also log information relevant to individual computers by IP address, should a machine be having problems for example.



For easier reading and sorting, the Alert Logs can be imported into Excel.

Alerts are often raised in the alert log to provide more information relating to alarms. When alarms occur its worth looking in the alert logs for more information in the event that an alarm is not self explanatory.

2.8 Steerpoints for Vessel and Mobiles

2.8.1 Vessel Steerpoints

SP1 is always on the vessel. The selection of SP1 determines the position on the vessel to which the displayed position represents and the position that is logged as 'SP1' in the data logging files. In a cable lay, SP1 would be the stern, being the last known point of the cable. For and ROV survey, the position of SP1 is likely to be the ROV launch and recovery point.

SP1 is calculated from the GPS antenna and the vessels CRP, through heights, x and y offsets, vessel heading and with pitch and roll angles applied. It stands to reason therefore that the further SP1 is from the antenna, the more error is likely to be introduced into the calculation of the SP1 position.

Therefore, the following should be taken into account to achieve minimal errors;

- The GPS antenna should be mounted as close as is practically possible to the SP1 position
- The gyro calibration must be accurate
- The pitch and roll sensor must have corrections applied and must be in the right sense!
- Offsets must be measured as accurately as possible

2.8.2 Mobile Steerpoints

SP2 and SP3 are predominantly used for Mobile Steerpoints. However, they can also be used for additional vessel Steerpoints if there is a requirement to monitor and log 2 or 3 positions on the vessel. If there are two ROVs in use, the Steerpoints would be configured;

ROV1 as SP2
ROV2 as SP3

SP2 and SP3 are calculated from the vessel CRP. As with SP1, there is considerable room for error if measurements and calibrations are not carried out accurately, especially where the HPR is concerned. The further the mobile is away from the CRP, the more error there will be in positioning.

When working with mobiles (and HPR) the following should be taken into account to achieve minimal errors;

- The GPS antenna should be mounted as close as is practically possible to the CRP position
- The gyro calibration must be accurate
- The pitch and roll sensor must have corrections applied and must be in the right sense!
- Offsets must be measured as accurately as possible
- The HPR must be calibrated
- VOS must be taken into account

2.9 Mobile Configuration

In the Mobiles configuration, there is a distinct difference between SP and Positioning in the Mobiles configuration table.

SP relates to the Survey Point of the ROV or the plough and it is the point on the vehicle that we are interested in.

Positioning relates to the selected HPR beacon on the mobile. It is very important that the offsets for the beacon agree with the beacon's actual position on the mobile. It is easy to select say B22 on the assumption that B22 is on the correct offset point! If these are not set up incorrectly, then there will be errors in the vehicle position.

As a precaution, it is recommended that the Mobile Offset name is the same as the beacon name. Blue Spider will attempt to match the selected 'Positioning' beacon with an offset with the same name.

This implies that it is good practice to always place the beacons at designated positions on the ROV or plough, for example;

B11	Always Port Fwd
B22	Always Stbd Fwd
B44	Always Port Aft

2.10 Blue Spider and HPR Calibrations

This chapter is not intended as complete calibration guide, it offers some useful verifications of the positioning systems.

2.10.1 The HPR System

The HPR system is a Range, Bearing and Height system. (x, y and z). By default, the reference point for the HPR positioning in Blue Spider is generally the vessel CRP.

Therefore the pole offsets (from CRP) are applied in the HPR system, so that when Blue Spider see's the x,y,z data, it references it to the vessel CRP.

When preparing for an HPR calibration, the following list should be helpful;

- It's the HPR system being calibrated, nothing else. Therefore all vessel rotations should be done around the pole, not the CRP.
- In an ideal world, the GPS antenna would be above the HPR pole for a calibration. There are various ways to apply offsets to the GPS, but the ULTIMATE way is to feed a GPS directly into the HPR and apply the GPS offsets into the HPR.
- Realistically, we generally feed a GPS string from Blue Spider into the HPR, which is either the real GPS position (with antenna offsets in the HPR) or the CRP position (0 offsets in the HPR).
- Regardless of the method chosen, there is a fundamental MUST to ensure that the calibration

configuration is correct. The HPR is a navigation computer, and, as such, it computes Easting and Northing for the CRP. When the correct GPS input is applied, the geodetics correctly set up in UTM, the offsets correctly measured, the same gyro and MRU inputs applied, the calculations of Easting and Northing in the HPR and in Blue Spider will be exactly the same, or at least to 0.02m.

5 Do not proceed with an HPR calibration unless this simple check has been passed!

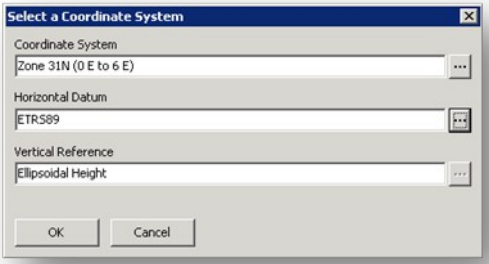
2.11 Geodetics

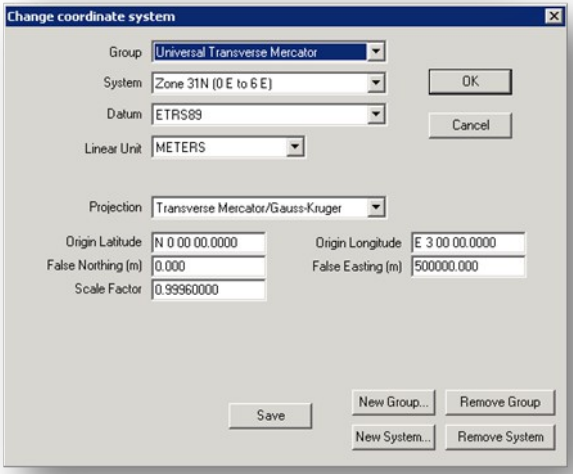
Blue Spider calculates all positions in Latitude and Longitude, and computes Easting and Northing values. The Geodetic settings in Blue Spider must be correctly applied for the current project. Heights depths and altitudes are now treated a lot more accurately in Blue Spider, to obtain sea levels (tidal influences), burial depths and cable altitude.

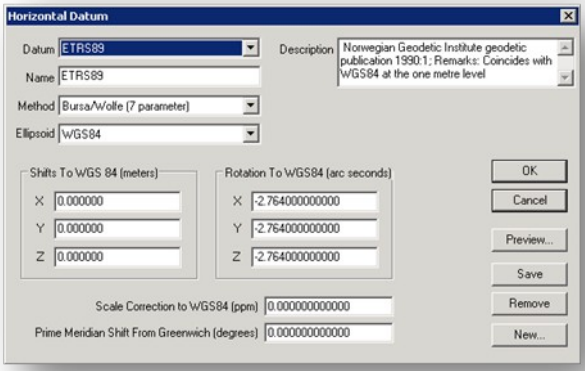
When configuring the geodetic information, the vertical Datum must be considered as well as the horizontal datum.

There are many BSPEngine variables to consider when checking the vertical offsets, including Water Line height, Draft, Geoidal Separation, and Altitude, use of the Variable Watch Windows is recommended during verification of these values prior to logging them.

The following screen shots show the new geodetic information panels in Blue Spider.


Select a Coordinate System
Coordinate System
Zone 31N (0 E to 6 E)
Horizontal Datum
ETRS89
Vertical Reference
Ellipsoidal Height
OK
Cancel


Change coordinate system
Group
Universal Transverse Mercator
System
Zone 31N (0 E to 6 E)
Datum
ETRS89
Linear Unit
METERS
OK
Cancel
Projection
Transverse Mercator/Gauss-Kruger
Origin Latitude
N 0 00 00.0000
Origin Longitude
E 3 00 00.0000
False Northing (m)
0.000
False Easting (m)
500000.000
Scale Factor
0.99960000
Save
New Group...
Remove Group
New System...
Remove System


Horizontal Datum
Datum
ETRS89
Description
Norwegian Geodetic Institute geodetic publication 1990:1; Remarks: Coincides with WGS84 at the one metre level
Name
ETRS89
Method
Bursa/Wolfe (7 parameter)
Ellipsoid
WGS84
Shifts To WGS 84 (meters)
X 0.000000
Y 0.000000
Z 0.000000
Rotation To WGS84 (arc seconds)
X -2.764000000000
Y -2.764000000000
Z -2.764000000000
Scale Correction to WGS84 (ppm)
0.000000000000
Prime Meridian Shift From Greenwich (degrees)
0.000000000000
OK
Cancel
Preview...
Save
Remove
New...

2.12 The BSPEngine.ini File

The BSPEngine.ini file is used to configure BSPEngine. The complexity of the INI file is a trade off for the flexibility it provides Blue Spider. The most common sections of the BSPEngine.ini file that might need to be configured are:

- System setting
- Cable Engine type
- GPS message format
- Heading message format
- Echo sounder message format
- Pitch and roll offsets
- Custom inputs for ROV / Plough, etc
- Custom outputs for ROV / Plough, Video overlays, etc
- Custom Variables that can have maths applied
- Logging section where CSV logs are formatted

The file is broken into sections and is documented along with examples in Appendix A (p.149)

The INI file can be edited in any text edior such as notepad. When changes are made to the INI file BSPEngine will notice the change and attempt to apply the configuration. If there are errors in the configuration alarms will be displayed and further information will be available in the alert log. This should enable the errors to be easily corrected.

2.12.1 BSPEngine.ini file Sections

Quick reference for finding documentation relating to configuration of the following

Category	Refer to
GPS Data	A.3.1.6 [Nav1]... (p.178)
Gyro Data	A.3.1.7 [Gyro1]... (p.181)
Echo Sounder	A.3.1.8 [Depth1]... (p.182)
Motion Sensor	A.3.1.9 [RP01]...[RP03] (p.182)
Cable Engines	A.3.1.2 [Plc]...(Cable engine configuration) (p.165)
Custom inputs	A.3.1.3 [CustomInputFormat1]… (p.166)
Custom outputs	A.3.1.5 [CustomOutputFormat1]… (p.171)
CSV Logging	A.3.1.15 [LogFile1]… (p.197)
SQL Logging	A.3.1.1 [System] (p.157)
Permissions	A.3.1.1 [System] (p.157) also A.4 Machine.acl.INI (p.206)
Other	A.3.1.1 [System] (p.157)

2.13 Custom Data Inputs

Decoding Custom Inputs into BSPEngine is split into two sections in the BSPEngine.ini file.

- Decoding the input string
- Assigning Variables to use in Blue Spider

This section covers how to decode strings. The next section will show how to create and assign variables.

2.13.1.1 String Formats

Most data is presented as csv (comma separated variables) and has a header.

$HEADER,Data1,Data2,Data3,Data4

MsgName is used to detect the header using the code. Use FieldType to specify a length for the header. (Default (0) can also be used to read the whole field.)

MsgType = 1,0,7 (1st field, Read 7 Chars)

$ H E A D E R
1 2 3 4 5 6 7

Because this string is comma separated, the middle decode (0) is not used.

Field1 = 2,0,0 (2nd field, read all chars) = Data1
Field2 = 3,0,0 (3nd field, read all chars) = Data2

In this example the string does not contain and comma separators, but the data arrives in fixed size blocks.

HEADERData1Data2Data3Data4
12345678901234567890123456 7

We can decode this as follows

MsgType	= 1,1,6 Start Field1, Char1, 6 Chars of data
Field1	= 1,7,5 Start Field1, Char7, 5 Chars of data
Field2	= 1,12,5 Start Field1, Char12, 5 Chars of data

If we don't need to use the header, there is no need to decode it. However, if there is more than one message on a port, then it would be necessary to decode the headers to BSPEngine can distinguish the messages. For example, a GPS receiver could output 3 messages, $GPGGA, $GPVTG and $GPZDA.

After decoding (or parsing) either of the above examples, then the data in the string is available as **Fields** to BSPEngine.

In the following example, there are three other considerations to be taken in to account.

- [CustomInputFormat#]
- CustomInputChannel#]
- Connecting the Format to the Channel.

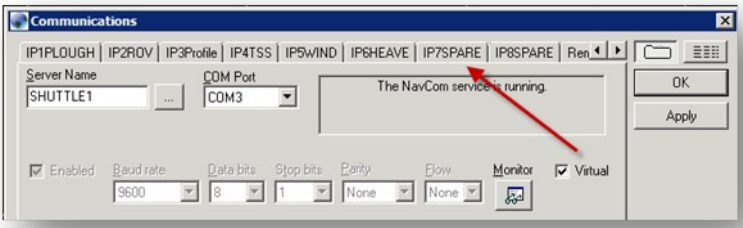
Every input format must have a unique number.
Every Input format must be linked to a channel.

It is possible (and likely) to have more than one input format assigned to an input channel. (For example, if decoding the GGA, ZDA and VTG formats of a GPS receiver.)

The Format and the Channel are connected with the Message line;

Message1 = CustomInputFormat1
(This is the only time the CustomInputFormat No is used)

The Input Channels relate to these Communications tabs.



```
[CustomInputFormat1]
MsgName                 = $PLOW
MsgType                 = 1,0,0
Field1                  = 2,0,0  ; Plough Pitch
Field2                  = 3,0,0  ; Plough Roll
Field3                  = 4,0,0  ; Plough Lay Cable Tension
Field4                  = 5,0,0  ; Plough Burial Depth
Field5                  = 6,0,0  ; Plough Water Depth
Field6                  = 7,0,0  ; Plough Steering Angle
Field7                  = 8,0,0  ; Plough Tow Length
Field8                  = 9,0,0  ; Plough Depressor Height
Field9                  = 10,0,0 ; Plough Tow Force
Field10                 = 11,0,0 ; Plough Heading
Field11                 = 12,0,0 ; Plough Trench Depth
Terminator              = "\r\n"
[CustomInputChannel1]
Message1                = CustomInputFormat1
```

More information relating to custom input decoding can be found in A.3.1.3 [CustomInputFormat1]… (p.166)

2.14 CSV Logging

Data can be logged to customised CSV files. The configuration for the logging file formats is in the BSPEngine.ini file. The CSV formats look like this.

[Logfile1]

Each log file format section must have a unique LogFile number.

Type = Event/Normal/Beacon

Event Only logged when a manual fix is taken.

Normal Logs all system created events.

Beacon Logs all beacons being received.

See the reference below for information on the other types of log files that are available.

{heading = "System Date"}

The heading defined here is what will appear in the logging files. Most of the BSPEngine variables use their existing defined names but these can be overridden.

For further detailed information refer to A.3.1.15 [LogFile1] ... (p.197)

```
[LogFile2]
Title   = ProjectLog
Type    = Normal
Field1  = Logging.FixNo
Field2  = Logging.EventNo
Field3  = System.Date  {heading = "System Date"}
Field4  = System.Time  {heading = "System Time"}
Field5  = SP1.Offset.Name {heading = "SP1 Offset"}
Field6  = SP1.Date {heading = "GPS Date"}
Field7 = SP1.Time {heading = "GPS Time"}
Field8 = Ship.Draft
```

```
[LogFile3]
Title   = Beacon
Type    = HPR
Field1  = Logging.FixNo
Field2  = System.Date  {heading =  "System Date"}
Field3  = System.Time  {heading =  "System Time"}
Field4  = SP1.Date     {heading =  "GPS Date"}
Field5  = SP1.Time     {heading =  "GPS Time"}
Field6  = Beacon.ID
Field7  = Beacon.X
Field8  = Beacon.Y
Field9 = Beacon.Z
Field10 = Beacon.Pos.Lat
Field11 = Beacon.Pos.Lon
```

2.15 SQL Logging

Database Logging works in conjunction with the Blue Spider Report Generator.

2.15.1 Brief Description

Every changing value generated by BSPEngine is stored in the SQL database if it is running! This is different to logging data at a fixed time interval. For example, GPS may be input into BSPEngine at 5Hz and Heading may be input at 10Hz. In a one second interval, SQL will have logged 15 messages for this GPS and gyro. Every message is stored with a timestamp, identification and value. This creates a lot of data, typically 150 – 200MB / hour.

When the Database Report Generator is run, it selects all the required data based on time or distance intervals between two fixed dates and times, and produces a csv style report.

2.15.2 Advantages of SQL Logging

- Retrieve only the data that is required.
- Templates can be used to produce reports.
- Fast access to relevant data.
- No risk of forgetting to record important data.

2.15.3 Limitations of SQL Logging

- There is no SQL backup logging in place yet.
- Data cant easily be read from multiple SQL files.
- Very large amounts of storage required.

2.15.4 What gets recorded

The data recorded in a SQL database is recorded to a number of different tables storing information falling into several categories. This data includes all raw input messages, most calculated variable values (any that are not recorded are ones that have equivalents that can be used to calculate), all alarms and alerts and optionally system performance information.

- Raw input messages
  This means we can playback all data at the rates at which it was received. If we made a mistake in configuring the system we can recover the situation retrospectively as a last resort. We also have a complete record if an incident should occur.
- Variables values as they change
  This means we can generate reports and construct CSV files from the logged data. The CSV log files generated this way can have any desired interval e.g 1 second, 5 seconds etc.
- Alarms
  We can correlate occurrence of alarms with changes in configuration, input variable values or arrival of unexpected data.

Certain variables are deliberately omitted from being recorded to the SQL database. For instance easting and northing values are never recorded (except when recording raw input messages) but latitude and longitude are always

recorded. Easting and Northing can be reconstructed by the report generator.

3 Surveyor Tasks

3.1 Route Lines Points and Targets

3.1.1 Route Lines

Route Lines can be manually entered but are generally imported from spreadsheets. In order to import from a spreadsheet, Microsoft Excel™ must be installed on the same computer as Blue Spider. Blue Spider will produce a warning otherwise.

3.1.2 Importing Routes

WARNING. Blue Spider assumes that all imported geodetic positions are in WGS84 regardless of the spheroid selected in Blue Spider at the time. Failure to observe this could easily lead to the route being misplaced if not working in WGS84.

Route Positions Lists issued by GM Charting Dept do not have any geodetic information on them. The geodetic information should be verified before importing the data.

3.1.3 Importing Transformations

Data is in WGS84

Blue Spider can be configured to be in WGS84 for the data import. Geodetic data can be imported.

After importing the data, the geodetic settings in Blue Spider can be changed to the working geodesy. The position data will be transformed accordingly.

Data is not in WGS84

Configure PPT for the same spheroid as the data set and import the data into PPT, then - either

Option a)

Change PPT to WGS84 and output the Latitude, Longitude and Depth coordinates, then import them into Blue Spider (set for WGS84).

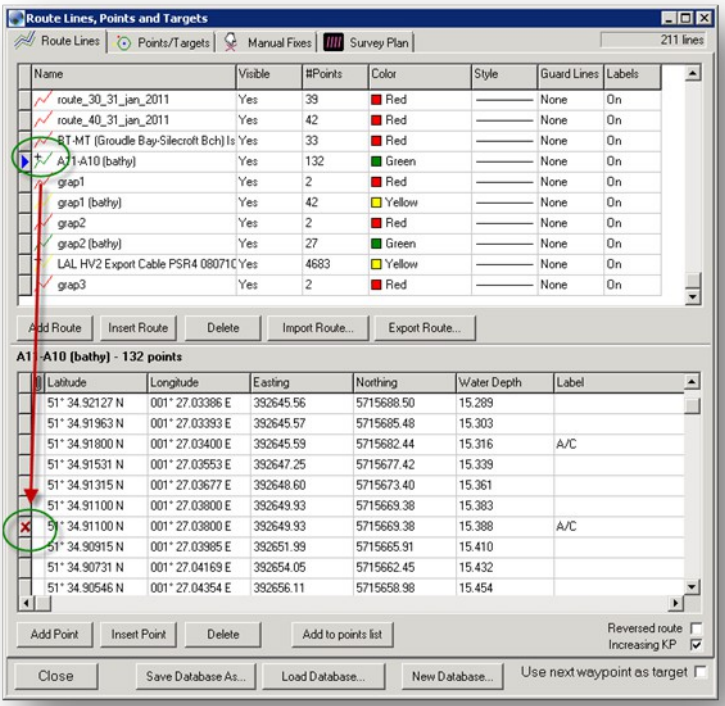
or

Option b)

Export the data in Easting, Northings and Depth and import directly into Blue Spider with the same geodetic settings. Blue Spider will assume that these positions are in the working geodesy.

3.1.4 Viewing Route Lines, Points and Targets

The top section of this dialog box shows the Routes (Tab), and the bottom section provides the details of the (selected) route, including bathy information.



The + sign to the right of the Route name is an indication that there are repeated points in the Route. These are not generally a problem but it is still worthwhile investigating the data and removing one or more of the duplicated positions.

3.1.4.1 Route Options

The bottom of the page shows two option select boxes, these are easily overlooked.

- Reverse Route
- Increasing KP

Using these options, it is possible to rearrange the points in the route to match the KPs in the RPL. The key here is that Blue Spider will never import KPs from any file, it will always calculate them. Therefore it is possible to import a Route from an Excel sheet only to find that the KP scale is the wrong way around.

Blue Spider also makes the assumption that the first coordinate imported relates to 0.000 KP. It may not be!

The value of the 1st KP in Blue Spider can be changed after the data is imported. All other values will automatically update.

After importing the Route data, the following checks should be carried out in Blue Spider.

- Does the 1st KP in the RPL (Route Position List) agree with the 1st KP in Blue Spider?
- Are the KP's incrementing in the same direction as on the RPL?
- Does the last KP calculated in Blue Spider agree with the last KP in the RPL?

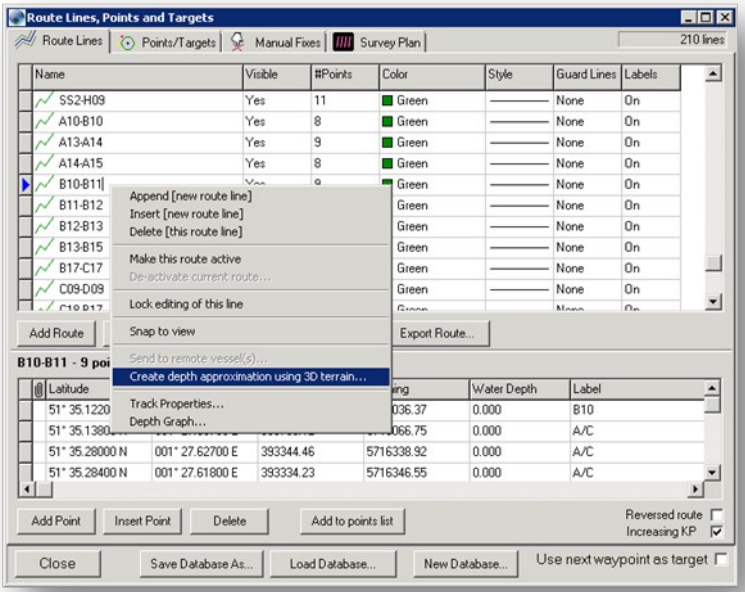
This last check could lead to finding issues in the RPL, as obviously the KP values that Blue Spider computes should agree very closely with the RPL. Errors are usually down to amendments having been made to the RPL incorrectly.

Once the route is imported, ALWAYS walk the route through the Blue Spider screen, looking for any obvious errors like spikes, kickbacks or obviously wrong AC (Alter Course) angles.

3.1.5 Creating Depths from Terrain Data

A new feature in Blue Spider allows a bathy line to be created from the terrain of a route line. The data is extracted from the 3D Viewer if terrain, (DTM) files are imported into the 3D Viewer.

The 3D viewer must be open at the same time as Blue Spider. Blue Spider will prompt with a warning if the 3D Viewer has not been started.



To use this feature, right click on a Route Line and select **Create depth approximation using 3D Terrain**

The user is given options to specify

- Maximum deviation from bathymetry
- Sampling step size

3.1.5.1 Maximum deviation from bathymetry

This relates to how far off the route line we are prepared to accept to achieve good data. The bathy data is saved as a series of points in a .PTS or an XYZ file. These are normally in a very simple format.

Easting, Northing, Depth

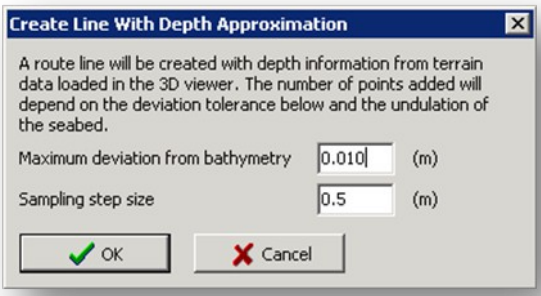
It is unlikely that any of these points will exactly match the points on the route line, so some deviation has to be acceptable, especially if there is not a lot of good quality bathy data.

3.1.5.2 Sampling Step Size

This tells Blue Spider how often (along the route) to look for depth data. If there is a large amount of good quality terrain data, say from a multibeam survey, then this option should be used with consideration to the total length of the route.

Sampling data at very close intervals will create a potentially massive route database (with depth) file which could lead to online problems if the file becomes to large.

In summary, the smaller the deviation and the step size, the more bathy points there will be in the route file. It is advisable to keep this to a sensible scale, depending on the nature of the work being carried out. It is unlikely for example that there would be a requirement for a depth reading every 10cm along the route!


Create Line With Depth Approximation
A route line will be created with depth information from terrain data loaded in the 3D viewer. The number of points added will depend on the deviation tolerance below and the undulation of the seabed.
Maximum deviation from bathymetry
0.010
(m)
Sampling step size
0.5
(m)
OK
Cancel

When the process has run, a new route line is created with Water Depth information included. New positions are interpolated for the new depths.

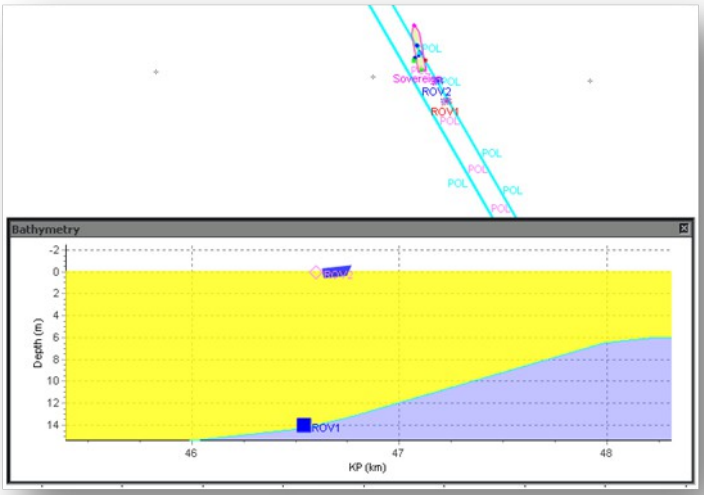
The new depth positions are not given labels.

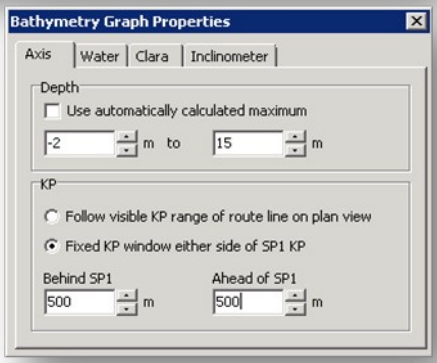
In Blue Spider this does not give cause for any problems, however, if the route data is exported from Blue Spider to a file, and re-imported back into Blue Spider, the points with no labels are ignored during the importing process.

The new Route Line can be made active and used in conjunction with a Depth Chart. Alternatively, the depth route can be used to display a Depth Chart window.

3.1.6 Depth Charts

Depth Charts can be shown on any Blue Spider screen by right clicking on the Depth route in the Route Lines box. Using the options, the vertical scale can be manually adjusted or left on automatic. The horizontal scale is determined by how much of the route is in view on the plan screen or by KP limits set either side of SP1.





The Depth Chart can be set to align itself with the plan view, so that at any orientation, the KP range that is 'in view' on the plan scale will also be 'in view' on the depth chart.

The Depth chart shows the following data against route KP;

- The vessel shape (distorted to maintain the scale)
- Any subsea vehicles
- HPR Beacons
- Seabed terrain
- Echo Sounder measured depth
- Sea Water Height (calculated from Veripos or RTK)

Additionally, the graph can display cable catenary; however this has not been fully tested and is by no means as comprehensive as say, MAKAI. This feature also depends on the availability of a cable database file (from MS Access) that holds data about the characteristics of the cable.

3.1.7 Creating Curved Routes

A point to point route can be given curves in Blue Spider. There are a few points to bear in mind when creating curves.

- A curved route will have a different KP length compared with the straight line route.
- All ACs can be given the same radius curve.
- Some ACs may need a different radius curve.
- Curved routes can be viewed but don't have to be the Active route.

In the **Route Lines, Points and Targets** page, right click on the route (to be curved) and select **Track Properties.**

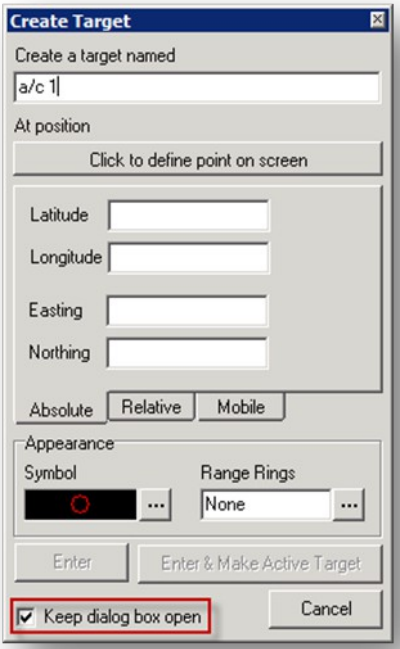
3.1.7.1 Route Track Properties

Track properties allow

- Modifying colour and thickness of the route.
- Modifying symbol colours and size.
- Curve radius options.
- Visibility of circles (used to create the curves).
- Option for all the curves to be the same or individual.

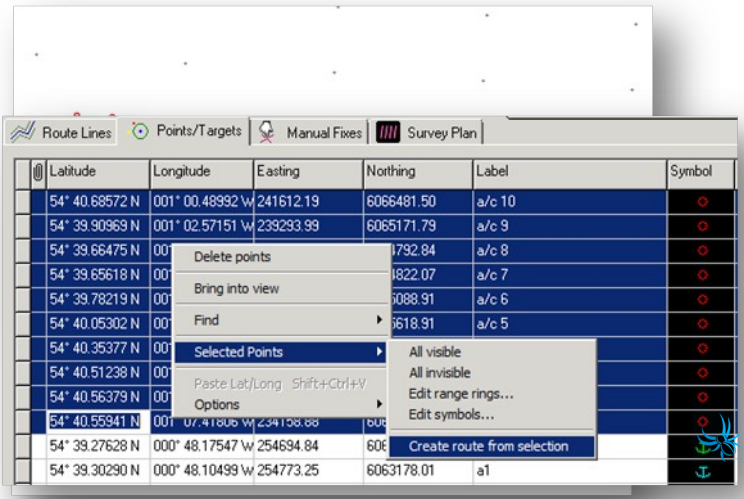
3.1.8 Creating Routes from Points

In this simple example, the Target button is to create 10 points on the screen. This is made easier by selecting the option to 'Keep the dialog box open'.



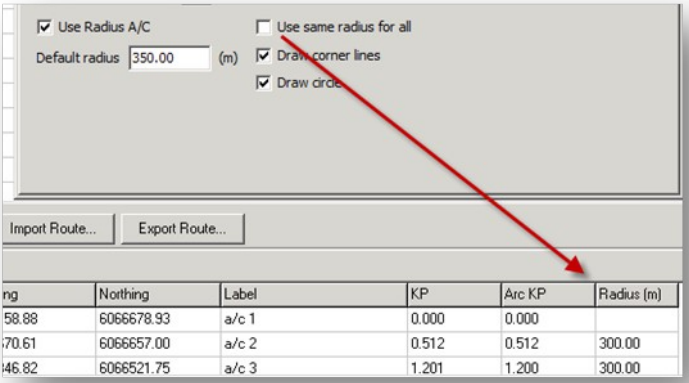
Using the **Click to define point on screen** and the **Enter** buttons, it is easy to create a series of points on the plan view.

The target names will increment as the points are created.



3.1.9 Creating Individual Radius Curves

When the curves are configured individully, then the radius is configured in the **Route Lines, Points and Targets** page.


Use Radius A/C
Use same radius for all
Default radius 350.00 (m)
Draw corner lines
Draw circle
Import Route...
Export Route...
ng | Northing | Label | KP | Arc KP | Radius (m)
58.88 | 6066678.93 | a/c 1 | 0.000 | 0.000 |
70.61 | 6066657.00 | a/c 2 | 0.512 | 0.512 | 300.00
46.82 | 6066521.75 | a/c 3 | 1.201 | 1.200 | 300.00

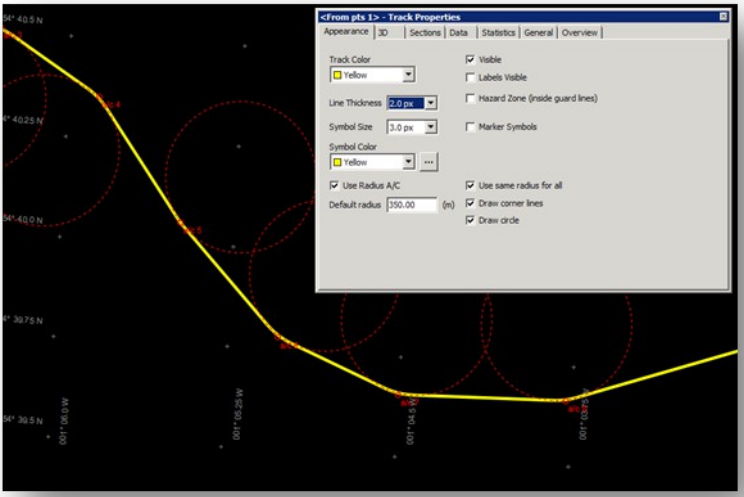
Note the difference in KP and the Radius Column below.

Label	KP	Arc KP	Radius (m)
a/c 1	0.000	0.000	
a/c 2	0.512	0.512	300.00
a/c 3	1.201	1.200	300.00
a/c 4	1.760	1.757	300.00
a/c 5	2.452	2.449	300.00
a/c 6	3.149	3.145	300.00
a/c 7	3.768	3.762	300.00
a/c 8	4.547	4.540	300.00
a/c 9	5.923	5.915	300.00
a/c 10	8.584	8.576	

3.1.10 Curve Radius Options

Once the **Track Properties** box is open the **Route Lines, Points and Targets** page can be closed. The Track properties will remain on the screen allowing the curves to be observed while they are being created.

The curves are created in real time while the adjustments are being made.



3.1.11 Grid, Altitude and Terrain Options

There are options in Routes to display less used options, these are

- Grid KP
- WGS84 Altitude
- Terrain Distances
- Guard Line Points

Clients who are members of the Flat Earth Society may require Blue Spider to log and display information in Grid.

WGS84 Altitude will show the GPS height of the seabed point from the WGS84 ellipsoid.

The Terrain distance is the distance along the ground, taking slopes into account.

KP	Grid KP	Terrain Dist (km)	Altitude WGS84
99.532	99.548	99.735	-48.547
99.541	99.557	99.744	-48.547
99.551	99.568	99.754	-48.548
99.561	99.577	99.763	-48.548
99.571	99.587	99.774	-48.549
99.580	99.597	99.783	-48.549
99.591	99.607	99.793	-48.549
99.600	99.617	99.803	-48.550
100.814	100.831	101.017	-48.595

3.1.12 Targets

Targets are not associated with routes, but they can be quickly created as positions on the screen or relative to a selected route. A target that is in use is said to be 'An Active Target'. Blue Spider can support up to 3 Active targets at any time.

When a target is active, Blue Spider can compute a range and bearing to the target.

3.1.12.1 Create Targets

There are 4 options available for creating Targets

- Absolute
- Relative
- Mobile
- Range

At anytime, the 'Click to define point on screen' button can be used to enter the position from the plan screen view.

Symbols and range rings can also be applied to the targets.

Targets are 'Entered' which adds them to the database, or 'Entered and made active' which also makes them the active target.

The 'Keep dialog box open' can be ticked to prevent the Target box from closing. This is a useful feature when generating multiple targets using mouse clicks.

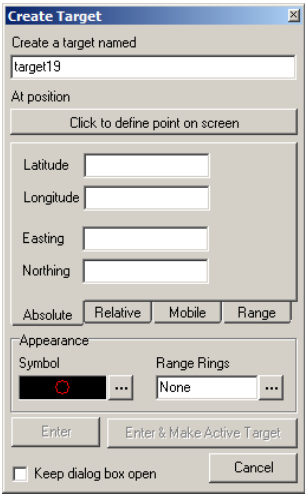
Absolute Targets are defined by position, in Grid or in Geographical.

Instantaneous conversion will be carried out as a position is entered. Relative Targets can be relative to

- Routes (Existing waypoints)
- KP and DOL from a Route
- Mobile Targets allow the user to select ROV or Plough as a target, this is a way to obtain a continuous range and bearing from the vessel to the selected mobile. The range and bearing represents a horizontal measurement, so slant range is not taken into account.
- Range targets (see page after next)

Target relative to waypoint on a route.

Range Targets are for creating targets relative to or from any point on the vessel that has a SteerPoint name. A Range Target might be used when attempting to maintain a fixed layback of a towed mobile from the stern for example.



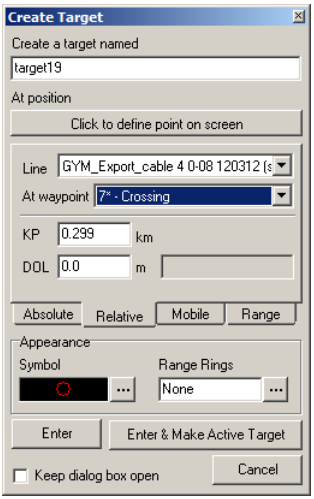
4 Barge Management System

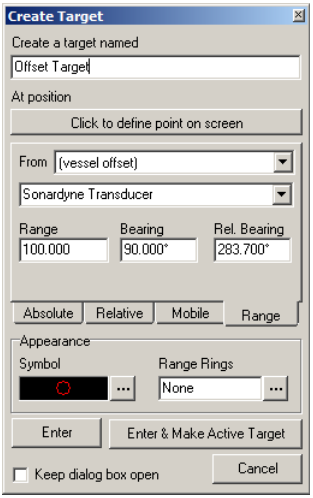
4.1 Overview

Anchor handling is built into Blue Spider. The system provides the following features.

1) One barge can work with multiple tugs.
2) An anchor position is selected on the barge and is automatically sent to the selected tug.
3) The tug confirms completion of the anchor drop or anchor recovery with a simple click, and the confirmation of the actual drop/recovery position is

received and logged on the barge.




Create Target
Create a target named
Offset Target
At position
Click to define point on screen
From (vessel offset)
Sonardyne Transducer
Range
Bearing
Rel. Bearing
100.000
90.000°
283.700°
Absolute
Relative
Mobile
Range
Appearance
Symbol
Range Rings
None
Enter
Enter & Make Active Target
Keep dialog box open
Cancel

4.2 Configuration for Barge Management

Blue Spider must be configured for Barge Management.

4.2.1 Equipment Required

- Wireless (Colubris) links between all the vessels or radio modem links (Satel).
- A single Blue Spider workstation on each of the tugs.
- A standard Blue Spider spread on the barge.

4.2.1.1 Colubris

Colubris is a TCP/IP based system that creates wireless network links from the barge to each tug. A wireless link is also established between each tug so that all the

vessels can see each other. Colubris uses WiFi technology. The closer the vessels are, the faster the network data link will be. The Colubris (one unit per vessel) has an IP address and becomes part of the Blue Spider network. Remote Desktop can be used over the Colubris links.

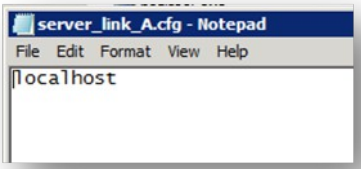
4.2.1.2 Radio Modem

Radio modems can be used to backup Colubris. Blue Spider does not have full functionality over a radio link but it can fully support the Barge Management System.

4.2.2 Software Configuration

4.2.2.1 BSPEngine Configuration

The barge and each tug must be running their own BSPEngine. In a typical configuration, the barge will run Box A and Box B, the usual Master/Slave arrangement. Each tug will run Box A only. This keeps the equipment on the tug to a minimum.



Note. ('localhost' can be used instead of the IP address). The default IP Address of localhost, or 'this machine' is 127.0.0.1.

4.2.2.2 Remote Vessel Configuration

A remote vessel configuration file must be configured in Blue Spider for Barge Management to work. The file contains the Vessel name and the name of the Computer (on that vessel) for all the tugs and the barge.

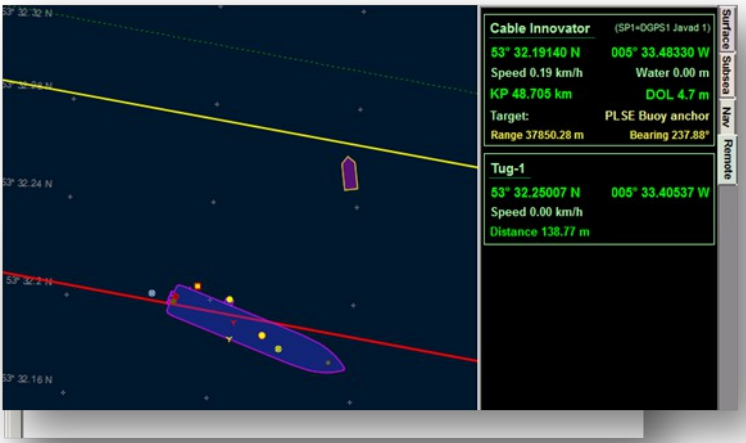
The file lives in the Blue Spider\System Config folder (on every vessel) and is called remote_vessels.conf

```
"Thomas" TugPC01
"Gordon" TugPc02
"James" TugPC03
"Fat Controller" OPS1
```

If this file does not exist, it can be created in Notepad.
Note. The computer name does not have any spaces.

4.2.2.3 Testing the Connection

Once the connection has been made, the other vessels will be seen on the Blue Spider screen. Using the Remote tab, the remote vessel details can be seen.



Remote vessels also appear in the Remote Positioning page in the Remote Vessels tab. Here the Name, Steerpoints and the IP address's of the remote vessels can be observed, but not altered.

4.2.3 Barge Management Features

4.2.3.1 Geodetics

The remote vessels will always share the same coordinate system settings. Changes made on one system will be automatically propagated to the other. Use the permissions (machine.ad.ini) file to limit geodetic changes to OPS1 and OPS2. On tugs it is generally desirable to prevent coordinate system changes altogether. For more information see A.3.1.1 [System] (p.157) also A.4 Machine.acl.INI (p.206)

4.2.3.2 Routes

By default, routes are not synchronised with remote vessels, this is because the route database files can become quite large. Use the Routes – Send to Remote vessels to send routes to all remote vessels.

4.2.3.3 AIS

A feature exists where, if the connection to the remote vessel is dropped, then the vessel shape can be used by the AIS position instead. The AIS must be interfaced to the Barge Server for this to function.

4.2.3.4 Restricting permissions

Permissions can be restricted to individual workstations so as to only allow anchor operations to be carried out from specific machines. For more information see A.3.1.1 [System] (p.157) also A.4 Machine.acl.INI (p.206)

4.2.3.5 Dropping own anchors

As well as the ability to request a tug to drop and recover anchors a barge can also drop and recover its own anchors.

4.3 Barge Management System Operation

4.3.1 Introduction

The Barge Management System uses a wireless network between barge and anchor handling vessels to allow passing of anchor targets from Barge to Tug and Anchor laid/recovered positions from Tug to barge. When more than one tug is assisting the Barge Master has full control over which anchor is assigned to which Tug. Anchor positions are logged at three different stages.

- Planned Deployment Position
- Actual Deployment Position
- Actual Recovered Position.

The reference point of the anchor tug used to derive the position of the anchor at deployment will be the centre point of the deployment roller at bow/stern of the anchor handling vessel (AHT), Steer Point 1. In the case of a Barge deploying their own anchors it is also possible to use the anchor handling function within Blue Spider to track the deployment of and recovery of the anchor pattern and in this case no tugs are used.

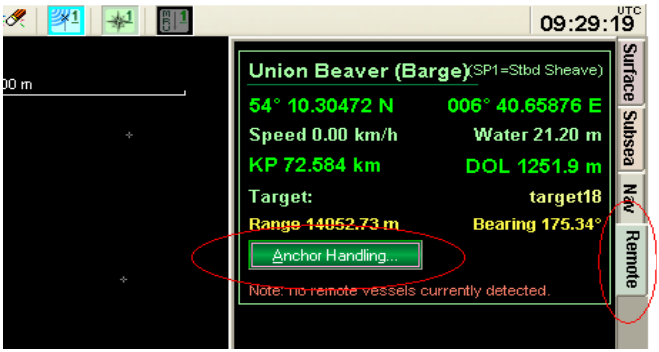
4.3.2 Safety

During cable operations it is of paramount importance to prevent contact between any seabed surface structures (pipe lines / cables/ seabed sensors/ UXO targets) and the anchors and anchor cables. To ensure this when an anchor has to be deployed from the barge over any seabed obstruction the anchor must be held on the deck of the anchor handling tug and the wire passed out from the barge under tension to suspend the wire above the seabed as the AHT moves towards the deployment point. Separation distance between barge and seabed structure should be minimised if at all possible as this assists separation distance between wire and structure. Both the Barge Master and Tug Master are responsible for ensuring that any known cables, pipelines and seabed structures are avoided. It is the responsibility of the Survey personnel to make sure that all known obstructions are displayed to the Barge and Tug Masters, they should also monitor anchor positioning and cross check any prospective anchor positions remain clear of cables/pipelines/seabed structures/ obstructions//UXO targets by the project specific clearance distance. Range rings or Guard Lines can be used around these positions to help maintain this distance.

THE ONLY OVERIDE TO THIS WILL BE VESSEL SAFETY WHEN IF NECESSARY TO SAFEGUARD PERSONNEL/VESSELS FROM HARM AN ANCHOR MAY BE DEPLOYED IMMEDIATELY TO THE SEABED. THIS WILL BE AT THE DISCRETION OF THE BARGE/TUG MASTER FOR WHOM THE ULTIMATE SAFETY OF THE VESSEL/PERSONNEL RESIDES.

4.3.3 Operation at Barge For Deployment

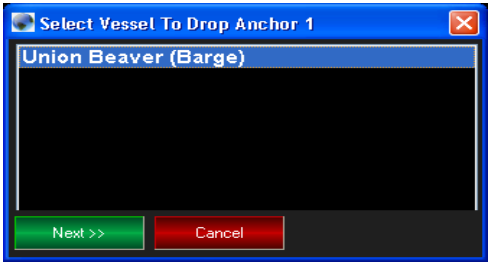
The system will automatically recognise the anchor handling tugs on the Blue Spider system and provide a notification when none can be detected. Access to the barge management system functions is obtained through the *Remote* tab. The process will be initiated at the Barge when the Barge Master will decide which



Remote Tab/ Anchor Handling Button

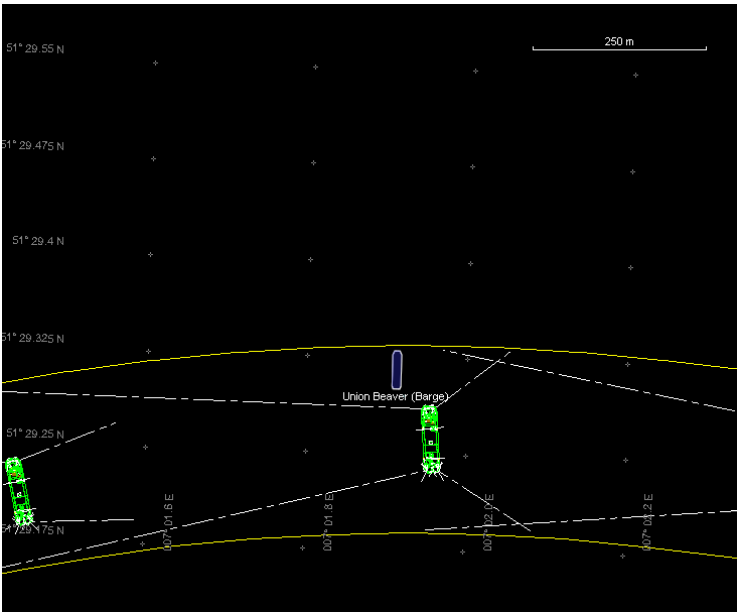
anchor will be deployed and the by which anchor handling vessel.

By clicking on the relevant anchor and then *Request Drop* the following Dialog is presented in which the required AHT can be chosen for this anchor operation, note that in the Figure 3 below only the Union Beaver is shown in the list as no other remote vessels are currently detected.



Select Vessel

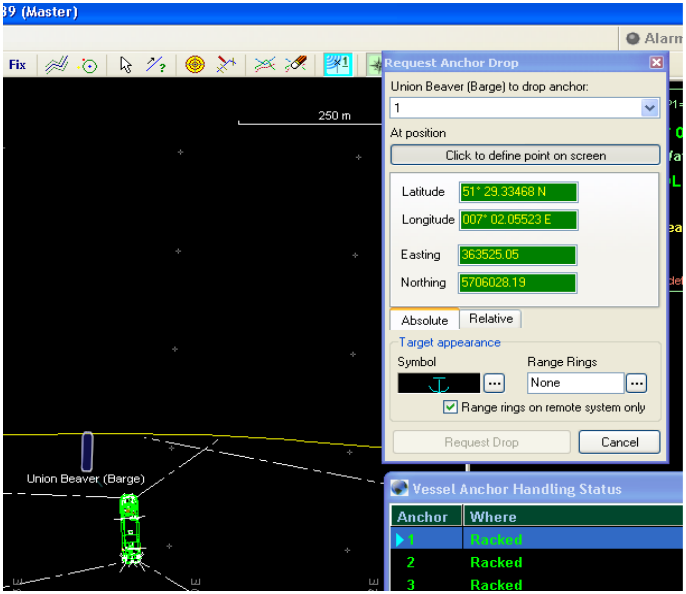
Once the AHT is selected and the *Next* button clicked you will then get the option to place the anchor target. ANCHOR TARGETS MAY BE PRE-PLANNED PRIOR TO THE PROJECT STARTING AND IN THIS CASE A TARGET POINT WILL ALREADY BE VISIBLE ON THE SCREEN. THIS MAY ALSO BE DISPLAYED AS A BACKGROUND DRAWING AS SHOWN BELOW.



Background of Proposed Anchor Patterns

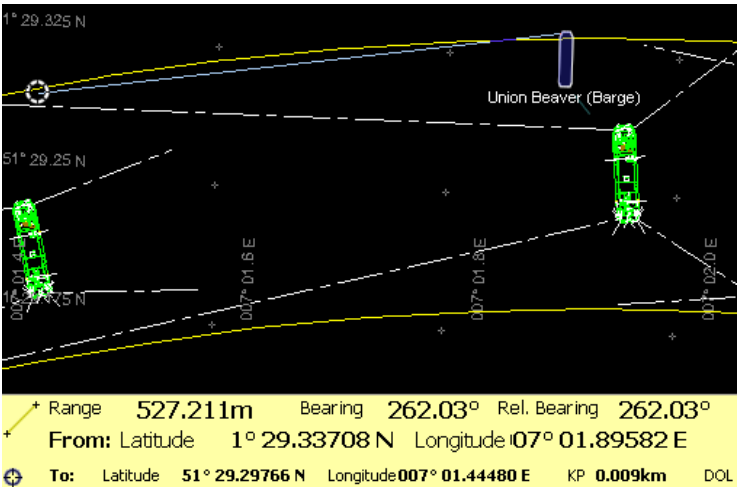
IF USING A BACKGROUND DRAWING SURVEY PERSONNEL MUST MAKE SURE THIS DRAWING IS LOADED TO THE BARGE MASTERS DISPLAY, TUG DISPLAY AND SURVEY DISPLAY. IT WILL NOT AUTOMATICALLY SHOW ON ALL SCREENS ONCE ENABLED ON THE MAIN SERVERS. ALSO REMEMBER THAT LARGE GRAPHIC FILES WILL PLACE AN EXTRA OVERHEAD ON THE SYSTEM. ONLY DISPLAY NECESSARY INFORMATION.

Anchor positioning can either be created by text entry of position or through placement of an anchor target using the cross hair cursor if the *Click to Define Point on Screen* button is clicked.



Request Anchor Drop

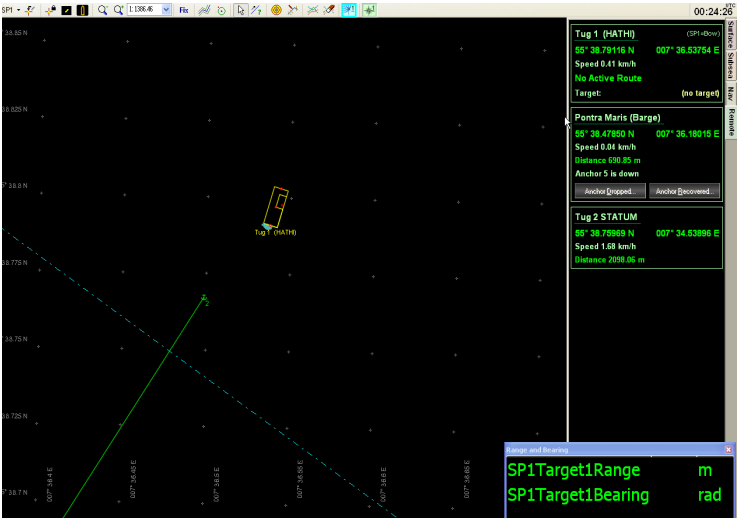
The figures shown above in green will update as the cursor is moved across the screen.


1° 29.325 N
Union Beaver (Barge)
51° 29.25 N
007° 01.4 E
007° 01.6 E
007° 01.8 E
007° 02.0 E
Range 527.211m Bearing 262.03° Rel. Bearing 262.03°
From: Latitude 1° 29.33708 N Longitude 007° 01.89582 E
To: Latitude 51° 29.29766 N Longitude 007° 01.44480 E KP 0.009km DOL

Target Placement Details

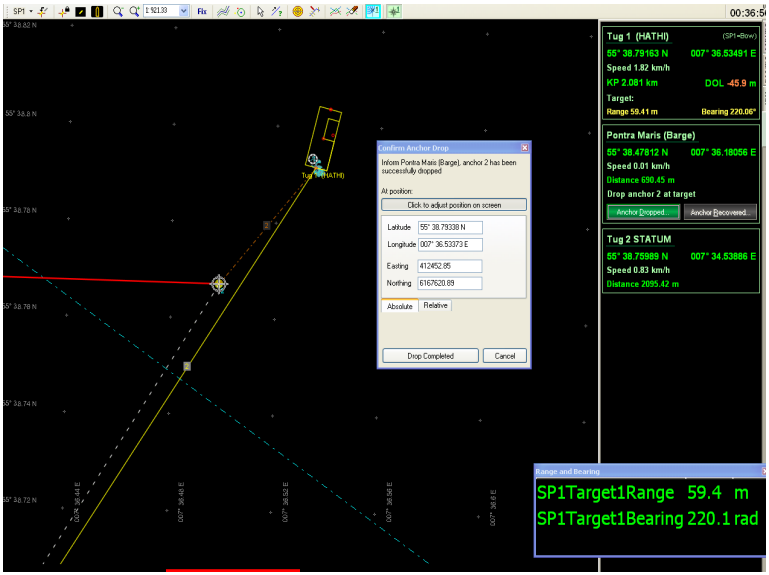
Once placed data in the form of Range and Bearing from the anchor fairlead will be displayed at the bottom of the screen. If it is found that the range/bearing is unsuitable then the target can be moved again merely by clicking at a different location. The Target will not be passed to the AHT until the *Request Drop Button* within the *Request Anchor Drop* dialog is pressed.

4.3.4 Operation at Tug for Deployment



Basic Tug Screen

Initially at the Tug both Anchor Dropped and Anchor Recovered are greyed out, this indicates that no operations are currently required by the AHT. Once an anchor operation is passed the relevant button will change to green.



Anchor Drop requested

Here anchor Number 2 has been selected as the target. The tug will follow the range and bearing to the drop location The tug can follow the brown dashed line to take it to the target as well.

Once the anchor has been dropped on the target the Confirm Anchor Drop window pops up. Click To Adjust Position on screen and move target cross hair pointer to the dropped position. Press the Drop Completed button. If the Drop Completed button is clicked without using the Click to Adjust Position on Screenthen the anchor recovery position is recorded as AHT Steer point 1 position.

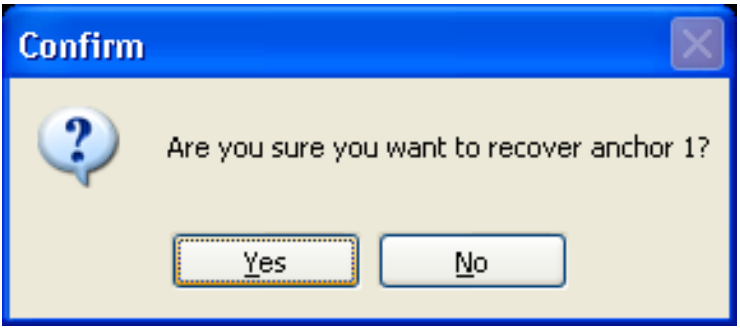
The anchor wire will now go green and update the dropped position both on the tugs and the barge. Note the anchor buttons are greyed out and no target is selected. The anchor line on the barge screen will update and the anchor status will show as dropped. Within the anchor log the positions of the planned deployment and actual deployment are recorded.

4.3.5 Operation at Barge for Recovery

For Anchor recovery the anchor to be recovered is selected at the barge within the Vessel Anchor Handling Status Dialog.

Vessel Anchor Status, Recovery operation

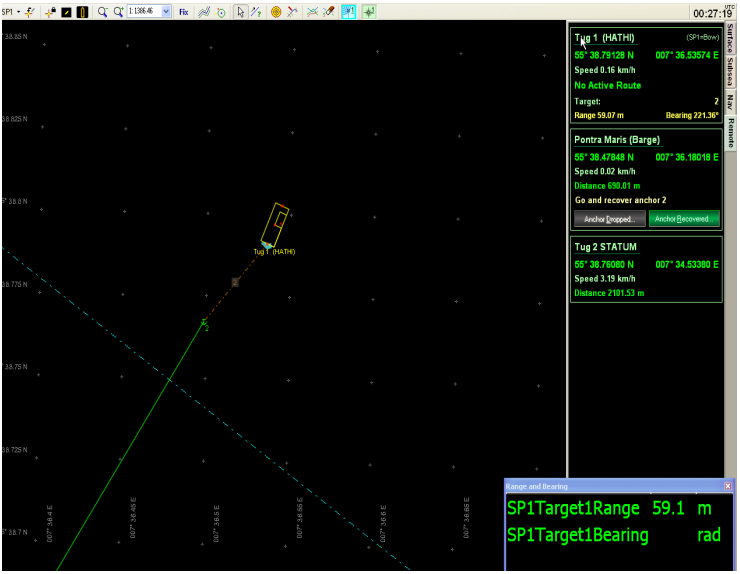
Request recovery by pressing the Request Recovery once the relevant anchor has been highlighted. At this point you will be presented a confirmation dialog.



Confirmation Dialog

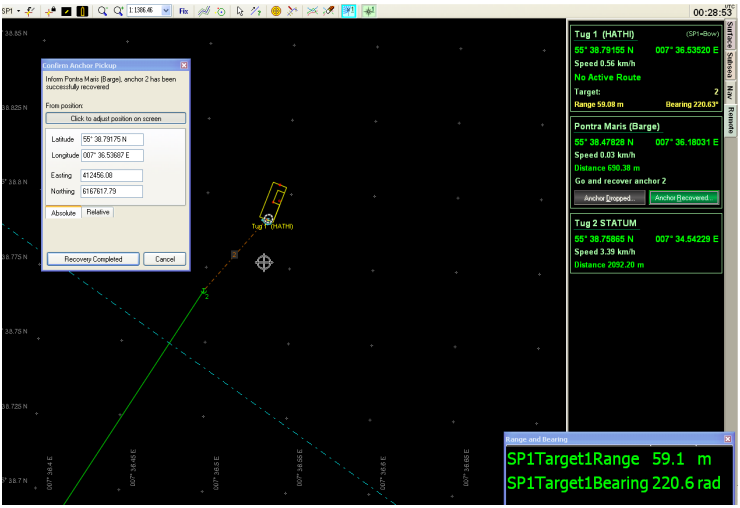
Once confirmed the anchor recovery operation will be highlighted on the AHT.

4.3.6 Anchor Recovery Operation at Tug



Tug Display, Anchor Recovery

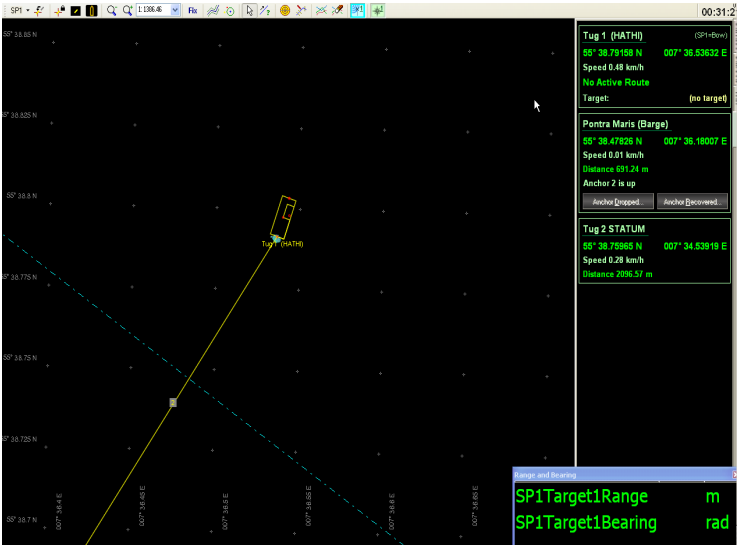
At the AHT the screen will display a heading line to show the direction of the anchor to be recovered. Once the tug has arrived at the anchor and commenced the pickup by pressing the *Anchor Recovered* button on screen the position at which the anchor was recovered can be recorded. This can be achieved by clicking on the *Recovery completed* button within the *Confirm Anchor Pickup* Dialog. If necessary it is possible to use the *Click to adjust Position on Screen* prior to pressing the confirm button if the actual position of recovery was missed due to the physical anchor operation at the tug.



Confirm Anchor Pickup

Once recovered the anchor recovery position is reported back to the barge and recorded in the anchor log. The anchor

wire display changes to yellow as shown in the previous figure.



Anchor Recovered

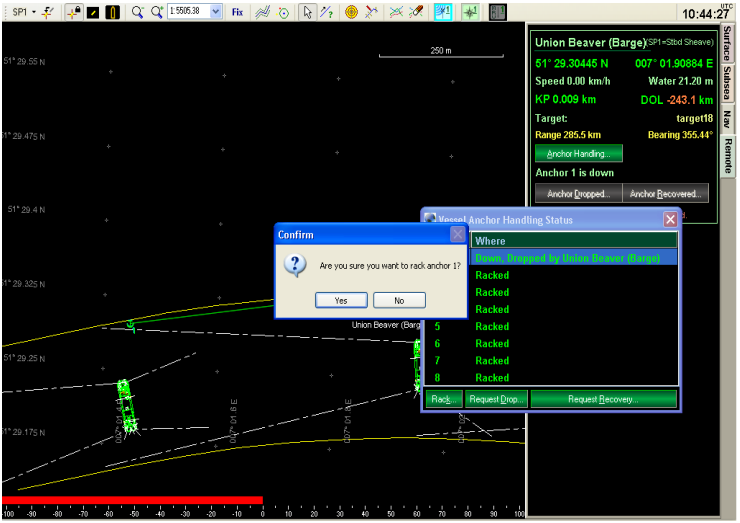
At this point a new target for deployment may be passed to the tug or the anchor may be racked.

4.3.7 Anchor Racking

To rack an anchor the barge selects the Rack option within the Vessel Anchor Handling Status Dialog.

Racking Option

A Confirmation dialog will then be presented to provide a level of security to this operation.



4.4 Anchor Logging

Anchor positions are recorded in a CSV format file for later processing if required by the project.

For more information refer to A.3.1.15.2 Type= (p.199)

4.5 Anchor Winch Support

Blue Spider supports bringing in data from anchor winches that have output messages to provide the payout length, speed and tension. In order to bring in such data we need to

define custom input messages and decode the results and assign to specific variables.

4.5.1 INI File Variables

You may define these input variables for each anchor

These are intended to be read from custom input messages

Anchor1.WinchTension	The winch tension in kN
Anchor1.PayoutLength	Length of the anchor wire in metres
Anchor1.PayoutSpeed	Payout speed in metres / sec

(and so on for Anchor2 etc)

4.5.1.1 BSPEngine Variables

If the vessel definition contains one or more anchors then BSPEngine will provide the following variables for each anchor:

Anchor1.Name	Name of the anchor as defined in the vessel definition
Anchor1.Status	A string “Racked”, “Up” or “Down” – Up means towed by a Tug
Anchor1.Name	Name of the anchor as defined in the vessel definition

Anchor1.Fairlead.Pos.Lat

Anchor1.Fairlead.Pos.Lon

Anchor1.Fairlead.Grid.Easting

Anchor1.Fairlead.Grid.Northing

Position of the offset on the barge to which the anchor line is connected.

(these are provided for convenience)

Anchor1.End.Pos.Lat

Anchor1.End.Pos.Lon

Anchor1.End.Grid.Easting

Anchor1.End.Grid.Northing

Drop of the anchor or SP position of tug if anchor is Up.

These are blank if the anchor is racked

Anchor1.Distance — Straight line distance to anchor

Anchor1.Bearing — Range and bearing to anchor end e.g. to tug or drop position. These are blank if the anchor is racked

(and so on for Anchor2 etc)

BSPEngine will also provide

Ship.NumAnchors

The number of anchors in the vessel definition

Ship.Anchors.Resultant.Force

Ship.Anchors.Resultant.Bearing

The resultant force vector (kN) of the combined WinchTension variables.

The existing AHT variables (intended for anchor logs) have been retained unchanged.

The new Anchor variables are primarily intended for use in displaying gauges and graphs.

5 Grid KP features

In Blue Spider (and BSPEngine) KP and DOL computations are carried out using rhumb line distances on the ellipsoidal surface of the earth. Less sophisticated survey systems and procedures may use grid distances instead. Some clients may insist on planning routes using grid distances. For this reason the ability to also calculate Grid KP and DOL has been implemented in Blue Spider (BSPEngine) and also PPT.

It is very important for surveyors and all other people involved to understand that (except for very small distances where differences are negligible) there will always be differences between the Grid KP and DOL and the True KP and DOL values. It is essentially as fundamental as the difference in length between a straight and a curved line!

Typically you try to compare Grid KP and DOL with the true values then you will notice differences. If a route line has two waypoints separated by some distance then in particular the grid DOL won't agree with the true DOL and will get worse as you move towards the midpoint between the two waypoints. The further the waypoints are apart the bigger the bust between Grid and True KP.

The reason for the difference is due to the way map projections work which is something that all surveyors should understand in great detail. You would probably use Grid KP with a UTM map projection (it would not be

at all sensible to use with Mercator except perhaps near the equator).

- Grid distances are NOT the same as distances on the surface of the earth.
- Grid lines are not straight lines on the surface of the earth.
- Rhumb lines are not straight lines on a UTM grid.

If you take a straight line on a grid and project it on to the surface of the earth it becomes curved and distorted. The effect of this depends on the position and orientation of the line. This means that a grid line may be longer than a rhumb line between the same two waypoints. The mid point of a rhumb line and a grid line are not likely to be at the same position.

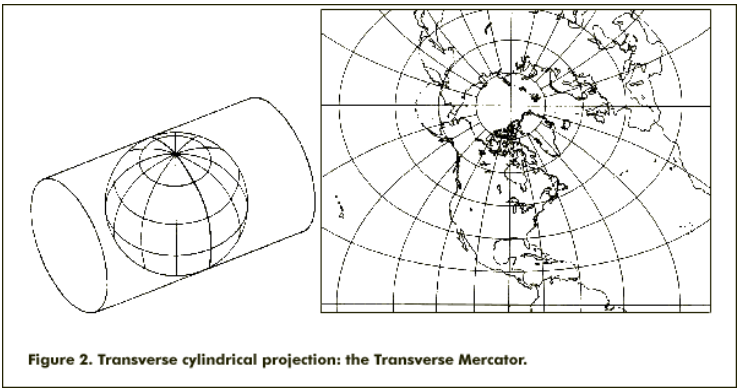


Figure 2. Transverse cylindrical projection: the Transverse Mercator.

Blue Spider always draws lines on the screen as rhumb lines but if you have loaded a background chart which contains a line that should be coincident with your route you can notice that the chart line is drawn as a

grid line and diverges from the route. Zoom in on the middle of the line and the chart and the route line may now be some distance apart.

5.1 Recommendation

When a client has requested that we work in Grid KP there is ONE step that SHOULD ALWAYS be taken in order to avoid confusion. Since Blue Spider always works in True KP and cable distance deviation etc. needs to work using true distances there is only one correct solution.

ADD EXTRA WAYPOINTS TO ROUTES DURING ROUTE PLANNING

Any long sections should be divided in half using the grid midpoint and an extra waypoint inserted at this position. This should be repeated until all sections are relatively short. In this way the route line now contains no long sections and (in UTM at least) the True KP and DOL should match the Grid values much more closely.

NAVSYSTEMS (IOM) LIMITED

BSPENGINE

INI FILE CONFIGURATION REFERENCE

Appendix A

BSPEngine Configuration Reference

Draft 1.0

A.1 Introduction

BSPEngine.INI defines much of the configuration data for Blue Spider and allows for decoding of standard and custom inputs. Definition of variables to be logged. Custom output formats and set-up of CSV log files

This appendix is intended as a guide to using the INI file for configuration and as reference material related to all aspects of configuration.

A.1.1 Install location

On a 32 bit machine the installation location will typically be:

```
C:\Program Files\NavSystems\Blue Spider
```

On a 64 bit machine the location will typically be

```
C:\Program Files (x86)\NavSystems\Blue Spider
```

A.1.2 Configuration data

Configuration data is not supplied by the install program and must first be created before the software can perform useful functions. However the location of the configuration data needs to be known.

A.1.2.1 Configuration location

On a 32 bit machine the configuration files typically be found under:

```
C:\Program Files\NavSystems\Blue Spider\System Config
```

On a 64 bit machine the location will typically be

```
C:\Program Files (x86)\NavSystems\Blue Spider\System Config
```

The System Config folder is where the initial **BSPEngine.ini** file must be placed.

Without this configuration file the service will start but will not perform any useful functions.

A.1.2.2 System Alert Logs location

The BSPEngine always records significant configuration events warnings and errors to a set of alert log files. The last few alert logs are always retained and older logs are automatically deleted. These built in logs require no further configuration. You can also set up your own alert log files.

The alert logs provide a means of checking that the system is working correctly and make it possible to see if there are any configuration errors.

The latest alert logs will be found in...

On a 32 bit machine the installation location will typically be:

```
C:\Program Files\NavSystems\Blue Spider\System Logs\Alerts
```

On a 64 bit machine the location will typically be

```
C:\Program Files (x86)\NavSystems\Blue Spider\System Logs\Alerts
```

A.2 Variables

BSPEngine works by assigning incoming data to variables and by logging these variables. These variables can be created in the INI file and can also be the result of arithmetic or other operations or in fact any valid Javascript expression. There are a number of built in variables are variables that are created automatically but you can also create your own variables.

Variable names use a dotted notation.

e.g.

Ship.WindSpeed

Variables can be used in expressions in the INI file such as

```
Ship.WindDirection = deg_offset(Ship.Heading,
            InputChannel5.Message1.Field1)
```

deg_offset() is a built in function for adding angles (in degrees) together (modulo 360). In this example InputChannel5.Message1.Field1 contains the wind direction from the anemometer but this is relative to the vessel

heading so to get the north relative wind direction we need to add two angles.

You can also add variables together with + or perform any other arithmetic.

Variable can be numbers or strings so you can also do things like concatenate two string variables. Variables can also be arrays or composite values.

If you need to perform a more complex transformation of an input variable you can write your own Javascript code and use it

For more information see the list of available built in variables and functions.

A.3 BSPEngine.INI

The BSPEngine.INI file contains much of the configuration required by Blue Spider. This is where you configure the inputs/outputs, variables, log files etc. The INI file is placed in the (A.1.2.1 Configuration location directory).

A.3.1 INI File Sections

The INI file can contain some or all of the following sections:

Section name	Purpose
[System]	Can specify the remote configuration location and a few other options.
[CustomInputFormat1]… [CustomInputFormat64]	Specifies decoders for custom input messages
[CustomInputChannel1]… [CustomInputChannel32]	Assigns message(s) (input formats) to individual comms channels.
[CustomOutputFormat1]… [CustomOutputFormat64]	Defines custom output formats and assigns these to given output channels (or even to log files).
[Nav1][Nav2][Nav3]	Specifies decoders for built in GPS decoder
[Gyro1][Gyro2][Gyro3]	Specifies decoders for built in Gyro decoder
[Depth1][Depth2][Depth3]	Specifies decoders for built in Echo sounder decoder
[RP01][RP02][RP03]	Specifies options for the built in motion sensor

Section name	Purpose
	decoder
RTT_01]...[RTT_08]	Sets up RTT input decoding
[Plc]	Cable engine configuration
[BodySchedule]	
[Variables]	Defines additional variables to be available for logging.
[VarHistory]	Allows history to be recorded for specified variables so that min, max, average and other statistics can be logged.
[LogFile1]…[LogFile16]	Defines individual log files, their types, layout, names, and recording strategy.

A.3.1 INI File Reference

A.3.1.1 [System]

The system section of the INI file can be used to set up a variety of global options.

A.3.1.1.1 DefaultInputTimeout=

The default input timeout specifies the amount of time for all custom inputs after which if no new message is received (for a given input message) the input variables will be set to blank.

You can override the input timeout for individual message formats associated with input channels.

If no input timeout is specified here or for an individual input then the input data variables will remain valid indefinitely.

It is often worthwhile specifying a default timeout as most input data has a limited useful lifetime. If no data is received on a particular input for a period of time it is usually best to log it as blank (e.g. unknown)

A.3.1.1.2 SpeedSmoothingInterval=

This is the interval of time over which the data for computing the ship speed should be sampled. If this key is not present the default value is 60 seconds. A value between 2 and 400 seconds can be specified here.

A.3.1.1.3 WaterlineSmoothingInterval=

This is the period of time over which the average waterline is calculated. The default value is 60 seconds. Any value between 2 and 800 seconds can be specified here. A longer value is recommended as the average waterline only changes slowly with the rise and fall of the tide.

A.3.1.1.4 MaxIntegratedNavTimeError=

This sets up filtering for integrated GPS mode.

This is the maximum allowed discrepancy between timestamps from different GPS receivers. If a greater difference is detected then providing 3 GPS receivers are configured the odd one out with the greatest time difference will be automatically rejected.

Any value between 0.5 and 10 seconds is allowable. The default value is 0.9 seconds.

A.3.1.1.5 MaxIntegratedNavDistanceError=

This sets up filtering for integrated GPS mode.

This is the maximum allowed discrepancy between positions from different GPS receivers. The distance difference here is the difference in the CRP position computed by each GPS receiver. If a greater difference is detected then providing 3 GPS receivers are configured the odd one out with the greatest distance discrepancy will be automatically rejected.

Any value between 0.1 and 400m is allowable here. The default value is 5m.

A.3.1.1.6 GPSAutoChangeoverDelay=

This sets a time delay on decisions to make automatic GPS changeovers. If automatic changeover were to occur instantly then this could lead to the possibility of rapid switching between two different GPS receivers. The change over delay helps to prevent this situation from arising.

The default value is 1.8 seconds. Any value between 1.2 and 10 seconds is permitted.

A.3.1.1.7 HPRPoleRotationCorrection=

This allows an angle to be applied to HPR positions to compensate for a slight rotation of the HPR pole. The default value is of course 0.

A.3.1.1.8 AdjustHPRPitchAndRoll=

If this value is specified as true then the HPR data will be compenstated for vessel pitch and roll. The default behaviour is NOT to do this as it is assumed to have already been done by the HPR system.

A.3.1.1.9 Rov1HPRAltitudeDisabled=

or **Rov2HPRAltitudeDisabled=**

Setting either of these values to true will prevent HPR Z data from being used to compute the altitude of a mobile. The default is to allow HPR Z to be used but a warning will still be displayed. The preferred method to compute altitude of a mobile is using bathy data from a pressure sensor on the vehicle. HPR Z is typically very inaccurate.

A.3.1.1.10 PloughHPRAltitudeDisabled=

This is the same as the ROV1HprAltitudeDisabled but for plough positioning.

A.3.1.1.11 PositionalSecrecy=

Positional secrecy is a feature that hides positional data on selected workstations. This is a feature that is typically used on sensitive installations where the position of seabed equipment must be kept a closely guarded secret. These are typically of a military nature. By default positions are displayed on all workstations. Setting this key to true will hide all position data on all workstations except for any specified in the Machine.acl.INI file. Refer to section A.4 (p.206)

A.3.1.1.12 AllowCoordSysChanges=

Setting this key to true prevents the coordinate system from being changed on all workstations. Nominated workstations with this permission can be set up in the Machine.acl.ini file. Refer to section A.4 (p.206)

A.3.1.1.13 AllowAnchorHandling=

Setting this key to true prevents anchor handling operations from being performed on all workstations. Nominated workstations with this permission can be set up in the Machine.acl.ini file. Refer to section A.4 (p.206)

A.3.1.1.14 MemoryUsageAlarmLimit

By default if BSPEngine memory usage exceeds 180Mb an alarm is raised. BSPEngine memory usage would typically be below this default limit but if large SDF files or route databases are used then its memory usage may be somewhat higher. This keys value sets the actual alarm limit (in bytes).

A.3.1.1.15 MemoryUsageSuicideLimit

By default BSPEngine memory usage can go to the maximum possible. As an option it is possible to tell BSPEngine to commit suicide and terminate if the memory usage goes above a threshold limit. The value here is specified in bytes. Setting a value of zero means that no limit is set and this is the default. This value should only be set in the unlikely event that there is suspected problem with the software and if specifically advised to do so.

A.3.1.1.16 StopSystemTimeGPSAdjustment

By default BSPEngine does NOT synchronize the computer system time to GPS time. You can set this key to 0 to enable the feature but it is not recommended. Synchronisation of the computer time to GPS time is not accurate enough to be worthwhile and can lead to unexpected behaviour.

Common sense would say that the computer time should be synchronized to GPS time but the GPS data is typically only received once a second and slight variations in latency and computer performance may cause slight variation in the system time if constantly being adjusted, so use initially and then disable the updating with a 1.

A.3.1.1.17 SQLServer

This specifies the machine name or IP address of an SQL server for use with the SQL logging feature. If the SQL server is running on the same machine as BSPEngine then specify localhost as the SQL server address.

A.3.1.1.18 SQLPort

The default value is 3306

A.3.1.1.19 SQLUserName

This is the user name for accessing the SQL server.

A.3.1.1.20 SQLPassword

This is the password for accessing the SQL server.

A.3.1.2 [Plc]...(Cable engine configuration)

```
[Plc] ;CWM 3 channel PLC
EncoderType                          =Standard
HardwiredPLCReset                    =0
Shaft1                               = 2,0,0
Shaft2                               = 3,0,0
Shaft3                               = 4,0,0
Tension1                             = 6,0,0
Tension2                             = 7,0,0
Tension3                             = 8,0,0
Engine1                              = "Not used"
Engine2                              = "LCE"
Engine3                              = "Drum"
```

This section sets the type of cable engine (or shaft encoder) and identifies the fields used for counters and tensions.

The Engine1,2 & 3 can be used to give the cable engines real names that will be seen in the NavSlack software and the Blue Spider variables.

A.3.1.3 [CustomInputFormat1]…

Custom input formats specify how to decode an ASCII input message and break it up into fields

Example:

```
;  Anemometer
;
[CustomInputFormat5]
MsgName          = $IIMWV
MsgType          = 1,0,6
Field1           = 2,0,0 ;Direction
Field2           = 3,0,0 ;
Field3           = 4,0,0 ;Speed M/S
Field4           = 5,0,0 ;
```

Custom input formats typically identify a message by name and by using more than one format for a given input channel it is possible to decode inputs where more than one message is being received.

If a message does not have a name it cannot be distinguished from other messages so in this particular case only a single

format should be used. Most messages can be identified by a unique name so this restriction is not normally a problem.

By default a message is assumed to have its fields delimited by a comma character. The message is also assumed to be terminated with a carriage return line feed pair “\r\n”

A.3.1.3.1 MsgName=

This key specifies the name at the beginning of the message. The value may be optionally enclosed in double quotes. If a double quote character is part of the message name it should be escaped by using \”

A.3.1.3.2 MsgType=

This optional key specifies the length and position of the message name.

If a message is terminated by an NMEA style checksum then add NMEA at the end of the MsgType value. This indicate that the last field should terminate before the *XX checksum.

See the section on field specifiers.

A.3.1.3.3 Field1=

For each field of a message a FieldN key should specify its position. See section A.3.1.3.6 Field specifiers (p.169)

A.3.1.3.4 Delimiter=

The default delimiter is the comma character.

You can specify a different delimiter

e.g. If a message has fields delimited by a colon character then use:

```
Delimiter = ":"
```

Alternatively if the message has different delimiters you can specify an ordered set of delimiters using the syntax

For example a message has 4 fields delimited by a colon and semicolon

```
Delimiter = [":",";",";"]
```

A.3.1.3.5 Terminator=

The default terminator is “\r\n”. You can override this by specifying a different terminator.

A.3.1.3.6 Field specifiers

The FieldN and MsgType keys take a value of the form of 3 numbers

These 3 numbers (a,b,c) specify

a) The field index 1 based

b) The actual character position in the message. By default this should be 0 in order to use the position determined by the delimiters

c) The actual length of the field. Again if this is left at 0 the entire field is taken up to the next delimiter.

If the delimiter is set to nothing then absolute positions must be used. This is the way to decode messages where all fields are fixed size and no delimiters are used.

For fixed layout messages without delimiters the delimiter should be specified as an empty string.

e.g.

```
Delimiter=""
```

This is because the default delimiter is a comma and unless you specify otherwise you will run into problems.

A fixed layout message is one where each field has a fixed width and the overall length of the message never changes. If

you want to decode fields from such messages the values for **b** and **c** must be specified.

- **a** is not used but its good practice to give it the field number.
- **b** must give the (one based) character position where the field
- **c** specifies the length of the field.

An example can be found under 2.13.1.1 String Formats (p.91)

A.3.1.4 [CustomInputChannel1]…

Custom input channel sections assign messages to a given custom input channel.

Custom input channels are ports with the device names IP_01 to IP_32.

Example:

```
[CustomInputChannel5]
Message1        = CustomInputFormat5
```

You can assign more than one message to a channel. Just add a Message2= and so on.

For each field in each message assigned to an input channel a variable with the name

InputChannelN.MessageN.FieldN

Will be created. These variables hold the raw input values of each field.

You may log them directly or assign to other variables first.

In addition a variable

InputChannelN.MessageN

is created for each message to hold the full decoded message

Another variable

InputChannelN.MessageN.Timestamp

is created to hold the last time the message was decoded.

A.3.1.4.1 Message1=

Keys Message1 to Message16 can be used to specify the message formats for each message to be received on the channel.

A.3.1.5 [CustomOutputFormat1]…

Custom output formats specify how output messages should be built and which channels they should be output on the output channels are the ports with the device names OP_01 to OP_16.

Example:

```
[CustomDataOutputFormat3]
CustomOutputChannel= 3
MsgName                              = $TESTOUTPUT
Field1                               = Train1.Node1.Tag1.PV
Field2                               = Train1.Node1.Tag1.ID
NMEA                                 = checksum
```

A.3.1.5.1 CustomOutputChannel=

This key specifies the output channels this message should be sent to.

You can if desired send a message to more than one output channel.

Example:

```
CustomOutputChannel= 3, 5
```

A.3.1.5.2 LogToFile=

You can log messages to log files that have the log file type set to the Output type.

This key specifies a log file by number (or more than one log file). Log files specified here must be of the correct type.

A.3.1.5.3 MsgName=

This key specifies the name of the message. You can place the value in double quotes if necessary. If no message name is required then you can omit this key

A.3.1.5.4 Field1=

This key specifies the variable (or expression) to be logged for each field.

A.3.1.5.5 Delimiter=

The default delimiter is the comma character.

You can specify a different delimiter

e.g. If a message has fields delimited by a colon character then use:

```
Delimiter = ":"
```

A.3.1.5.6 Terminator=

The default terminator is “\r\n”. You can override this by specifying a different terminator. The terminator key is optional but if not present the default \r\n will be used. If you want to omit the terminator entirely then specify

```
Terminator = ""
```

A.3.1.5.7 NMEA=checksum

If this key and value are present an NMEA style checksum is added to the end of the message

A.3.1.5.8 WhenTimeout=

WhenTimeout can be used to specify an interval between successive outputs of the message.

Example:

```
WhenTimeout = interval(5.0)
```

The value must be enclosed in interval()

The value specifies a period in seconds between each output of the message.

If neither WhenTimeout or Trigger are specified the message will be output at the default rate of once per second.

A.3.1.5.9 Trigger=

As an alternative to regular output intervals it is possible to trigger an output to occur on arrival of a specific input message or variable update.

You can actually specify more than one trigger condition

Valid trigger names are of the format

- InputChannel1.Message1
  An input channel message

- PortInput.DeviceName
  A raw comms input channel where device name is a valid device name

- Any valid variable name
  Any valid variable name can be used. When the variable is updated then this will trigger output.

You can have multiple trigger conditions for the output of a message.

Example:

```
Trigger=InputChannel2.Message3, PortInput.GYRO1
```

When using port device variable names such as PortInput.GYRO1 in field specifiers the Trigger should also specify the same variable name.

If neither WhenTimeout or Trigger are specified the message will be output at the default rate of once per second.

A.3.1.6 [Nav1]...

Configures the built in GPS decoder

Example:

```
[Nav1]
MsgName                 = $GPGGA
MsgType                 =  1,0,6
Time                    =  2,0,0 ; ->GPS1.Time (if no GGA)
Latitude                =  3,0,0 ; ->GPS1.Pos.Lat
LatitudeChar            =  4,0,0
Longitude               =  5,0,0 ; ->GPS1.Pos.Lat
LongitudeChar           =  6,0,0
GpsQuality              =  7,0,0 ; ->GPS1.Quality
NumSatellites           =  8,0,0 ; ->GPS1.NumSatellites
HorizontalDilution      =  9,0,0 ; ->GPS1.HDOP
Altitude                = 10,0,0 ; ->GPS1.Pos.Alt
GeoidalSeparation       = 12,0,0 ; ->GPS1.GeoidalSep
DGPSAge                 = 14,0,0

[Nav2]
MsgName                 = $GPVTG
MsgType                 = 1,0,6
Heading                 = 2,0,0   ; CMG from gps
SpeedKmh                = 8,0,0   ; speed by gps

[Nav3]
MsgName                 = $GPZDA
MsgType                 = 1,0,0
Time                    = 2,0,0  ; ->GPS1.Time
Day                      = 3,0,0
Month                   = 4,0,0
Year                    = 5,0,0
```

These sections define the GPS messages that can be decoded. The values are decoded to built in variables and are then accessible for logging. You can have up to 3 GPS receivers but they must all use the same formats. Note that the number of sections in the INI file here refers to individual GPS messages and not the number of receivers.

Data to be decoded by the built in GPS decoder must arrive on ports GPS1 to GPS3 and will be decoded to the built in GPS variables.

The [Nav1] refers to the GPS format number. Anticipated formats from different GPS receivers include:

- $GPGGA
- $GNGGA
- $INGGA

Nothing else is likely to need changing in this section, however note may be taken of the Altitude and the Geoidal Separation fields, as these can vary in usage between GPS receivers. (GPS receivers output either Altitude and Geoidal separation or MSL Height). (Blue Spider usually takes care of this itself).

```
[Nav4]
MsgName         = $GPZDA
MsgType         = 1,0,6
Time            = 2,0,0
Day             = 3,0,0
Month           = 4,0,0
Year            = 5,0,0

[Nav5]
MsgName         = $GPVTG
MsgType         = 1,0,6
Heading         = 2,0,0
SpeedKmh        = 8,0,0
```

$GPZDA is the Time and Date message from the GPS receiver.

$GPVTG is the GPS speed message. If present, Blue Spider can use this speed as the vessel speed, rather than calculate a speed from changing position.

A.3.1.7 [Gyro1]...

Configures the built in Gyro decoder

Example:

```
[Gyro2]
MsgName         = $HEHDT
MsgType         = 1,0,6
Heading         = 2,0,0
```

Data to be decoded by the built in Gyro decoder must arrive on ports GYRO1 to GYRO3 and will be decoded to the built in GYRO variables.

Note that in this section, the term [Gyro1] refers to the format number, not the gyro number. There may be 3 gyros interfaced into BSPEngine, and if they are all $HEHDT format, then only one gyro format is required.

A.3.1.8 [Depth1]...

```
[Depth1]
MsgName                         = $SDDB
MsgType                         = 1,0,5
WaterDepthMetres                = 4,0,0
```

The depth message is likely to have to be changed to suit the echo sounder in use. Typical messages include:

- $DPDPT
- $DPDPS

As well as [Depth1] you can also have sections [Depth2] and [Depth3] for configuring additional echo sounders. In the vessel definition an offset should be defined for each echo sounder. Depths from each echo sounder can be displayed in watch windows or in the depth graph.

A.3.1.9 [RP01]...[RP03]

This section sets up options for the decoding of motion sensor data.

```
[RP01]
InvertRoll      = 0
InvertPitch     = 0
PitchOffset     = 0.0
RollOffset      = 0.00

[RP02]
InvertRoll      = 0
InvertPitch     = 0
PitchOffset     = 0.00
RollOffset      = 0.00
SwapPitchRoll   = 0
```

A.3.1.10 RTT Inputs [RTT_01]...

RTT inputs are general purpose inputs but are designed to take in positioning data in a variety of possible formats. There are currently 8 available RTT input channels.

There are several modes that each channel can be configured for

- radio_master
  used for radio modem connection to remote vessels

- radio_slave
  used for radio modem connection to remote vessels

- remote_hpr
  used for taking in HPR data from a remote vessel

- grid_input
  for positioning where the position is supplied as easting/northing

- pos_input
  general purpose positioning input

- multi_pos_input
  multiple target positioning similar to HPR

The RTT channels are each configured using a section in the INI file

e.g.

```
[RTT_01]
Mode=radio_master
```

Each RTT channel that is configured must have the Mode= set to one of the modes listed above.

A.3.1.10.1 radio_master/radio_slave=

This mode is used to provide a communication channel between vessels (typically for barge management). The barge has a radio modem for each tug and this is (typically) set to radio_master on the barge and radio_slave on the tug. It does not matter which RTT channel is used.

e.g.

Barge INI

```
[RTT_05]
Mode=radio_master
```

Tug INI

```
[RTT_01]
Mode=radio_slave
```

Is fine as long as RTT chanel 5 on the barge is connected to a modem that uses the same frequency as the radio modem on the tug

connected to its RTT channel 1. The only important rule is that every radio_master is connected to a single radio_slave. It is a point to point connection.

A.3.1.10.2 remote_hpr=

This rather unusual mode of configuration is rarely used. Essentially what it does is takes in HPR data originating from a remote vessel and injects the data as if it were local HPR data. It is used in the situation where a remote vessel has better or more accurate provision for receiving good quality HPR data. It is possible to use local HPR and remote HPR at the same time as long as the beacon identifiers are unique.

Example:

```
[RTT_02]
Mode=remote_hpr
Vessel="The Survey Vessel"
```

The remote vessel is identified by name and must be configured as a remote vessel and operating over the wifi and/or radio modem link. The RTT channel communication port is configured to take BSPNet data directly from a BSPNet machine on the remote vessel.

A.3.1.10.3 grid_input=

This mode is used whenever data in grid format (easting/northing) is used. It is otherwise very similar to the pos_input mode.

A.3.1.10.4 pos_input=

This is the general purpose versatile positioning input mode. It can take in a position as latitude, longitude or calculate a position from range and bearing or from relative DX, DY.

There are a number of additional options here

- VesselRelative

Setting VesselRelative=1 is used in the case of a range and bearing or DX, DY modes to indicate that the vector is relative to the vessel rather than north relative.

- FromOffset

In the case of range and breaing or DX,DY the vector may be from a specific vessel offset. Setting FromOffset="Stern Sheave" tells BSPEngine that the position is to be calculated relative to a specific vessel offset.

- ConvertFromDatum

In the case of latitude and longitude this option can be used to specify the input datum (it is otherwise

assumed to be WGS84) Contact support for more details if this option is to be used.

- InputTimeout

Applies to all input modes except the radio modem modes. Can be used to specify a timeout after which any custom fields are set to blank.

A.3.1.10.5 multi_pos_input

The multi_pos_input mode is exactly the same as the pos_input mode except it allows for positioning of multiple objects. The data received on a multi_pos_input RTT channel is injected as HPR data so HPR beacons are used for positioning of mobiles rather than the RTT channel itself. It supports the additional concept of the received position data having a unique id that gets mapped to a HPR beacon id.

A.3.1.10.6 All modes (except radio)

All the input modes can decode one or more messages Specified by Message1=, Message2= etc. The radio modem modes have no additional configuration that can be specified.

A.3.1.11 Examples

A.3.1.11.1 Phinns decode (pos_input)

```
[RTT_01]
Mode              = pos_input
Message1          = RTT_01_GPS
Message2          = RTT_01_WD
Message3          = RTT_01_HDG

[RTT_01_GPS]
MsgName           = $PIXSE,POSITI
MsgType           = 1,0,13
LatDegrees        = 2,0,0
LonDegrees        = 3,0,0
Altitude          = 4,0,0

[RTT_01_WD]
MsgName           = $PIXSE,DEPIN_
MsgType           = 1,0,13
WaterDepth        = 3,0,0

[RTT_01_HDG]
MsgName           = $HEHDT
MsgType           = 1,0,6
Heading           = 2,0,0
```

The [RTT_01] section defines the input mode for RTT channel #1 as a **pos_input**. Three different messages are defined: one for GPS, one for Gyro and one for Water Depth. The decode formats for each of these is defined in its own separate section and these are referenced by name in Message1, Message2, Message3. For the pos_input type each message can contain any field types supported by either the standard GPS, Gyro or Depth inputs e.g the same as [Nav1], [Gyro1] or [Depth1]. In addition custom fields such as Field1, Field2 etc.

can be specified. These if present will be available as variable RTT_01.Message1.Field1 and so on.

A **pos_input** type RTT channel will typically be used to position a mobile using something that provides position information in a way similar to the information provided by a GPS receiver e.g the input message will typically provide a position as latitude and longitude.

It is also possible to position using a relative Range and Bearing from a specific vessel offset in which case the system will compute the latitude and longitude.

An example of this is given, for the **multi_pos_input** type) but this method can also be used for single position input.

As an alternative to Range and Bearing, or absolute Latitude and Longitude it is possible to use DX, DY, (and optionally DZ).

A.3.1.11.2 Blueview (multi_pos_input)

```
[RTT_01]
Mode = multi_pos_input
VesselRelative = 1
AngleOffset = 0
FromOffset = CRP
Message1 = RTT_01_BlueView
IDFilter = 1,2

[RTT_01_BlueView]
Delimiter = [":", ",", ",", ","]
Terminator = "\r\n"
Field1 = 1,0,0
ID = 1,0,-2 ; take last two characters, ignore the rest
Range = 2,0,0
Bearing = 3,0,0
Field2 = 4,0,0
Field3 = 5,0,0
```

A.3.1.11.3 Fanbeam (multi_pos_input)

```
[RTT_02]
Mode = multi_pos_input
VesselRelative = 1
AngleOffset = 180
FromOffset = Roller
Message1 = RTT_02_FanBeam
IDFilter = 1,2

[RTT_02_FanBeam]
Terminator = "\r\n"
Field1 = 1,0,0
ID = 1,0,2
Range = 2,0,0
Bearing = 3,0,0
Field2 = 4,0,0
Field3 = 5,0,0
```

A.3.1.11.4 Gps on a plough (pos_input)

```
[RTT_01]
Mode        = pos_input
Message1    = RTT_01_GPS

[RTT_01_GPS]
MsgName                                 = $GPGGA
MsgType                                 = 1,0,6
Time                                    = 2,0,0
Latitude                                = 3,0,0
LatitudeChar                            = 4,0,0
Longitude                               = 5,0,0
LongitudeChar                           = 6,0,0
GpsQuality                              = 7,0,0
NumSatellites                           = 8,0,0
HorizontalDilution                      = 9,0,0
Altitude                                = 10,0,0
GeoidalSeparation                       = 12,0,0
DGPSAge                                 = 14,0,0
```

A.3.1.12 [ScriptIncludes]

Script includes allow you to create your own functions in Javascript. By creating a .js file and placing it in the System Config\Scripts folder and adding it as an include in this ini file section any functions you define will be available to the system.

You will need to initially create the scripts folder.

Example:

```
[ScriptIncludes]
Include1 = myfunctions.js
```

Script functions can access any variable defined in the INI file or built in variables. Variables can be passed as parameters to script functions defined in an external .js file.

You need a basic understanding of Javascript in order to write your own script functions. It is possible to write script functions that will have a detrimental affect on the performance of BSPEngine or worse still cause it to crash or lock up. Care must therefore be taken when adding and using your own script functions. You should of course verify the correct operation of scripts before using on vessel operations.

A.3.1.13 [Variables]

The variables section allows you to create variables from other variables or extract internal values from variables.

You create variables by specifying a name for the variable as the key and any valid Javascript expression for the value.

Example:

```
Ship.WindDirection = deg_offset(Ship.Heading,
InputChannel5.Message1.Field1) {format="%.2f" heading =
"Wind Direction"}

Ship.WindSpeed     = InputChannel5.Message1.Field3
{heading = "Wind Speed"}

Ship.AvgGPS.Pos = latlon(historyOf(GPS1.Pos.Lat).avg,
historyOf(GPS1.Pos.Lon).avg)
```

NOTE

Any variable you create in this section must be defined on a single line in the INI file. In the example above some of the lines have been wrapped.

There are a number of built in variables and these are documented in section A.6 Built in Variables

You can assign to limited number of certain built in variables here.

You cannot define the same variable more than once.

Variables are accessible from Javascript code as values but they also have other properties and these can be accessed using the special functions described in section A.8.2 Special functions

Variables can have additional attributes that may be defined in the INI file such as the heading (caption for log file columns), format specifiers and a number of other attributes. Attributes are defined in curly brackets after the definition.

User defined variables are recalculated when any of the variables on the RHS of the expression change. You can override this default behavior using the attributes.

For details on configuring variable attributes see section A.7 Variable attributes.

A.3.1.14 [VarHistory]

The VarHistory section makes it possible to record the historical values of variables to an internal buffer. This in turn makes the historical data available to scripting and makes it possible to calculate statistics such as the average value, minimum and maximum etc.

Example

```
[VarHistory]
GPS1.Pos = {seconds=60 rate_hz=1}
Ship.Heading = {seconds=120 rate_hz=1}
```

In the above example GPS1.Pos actually refers to all variables starting with GPS1.Pos. GPS1.Pos will normally at least have GPS1.Pos.Lat and GPS1,Pos.Lon members.

When adding a variable like GPS1.Pos (which is a composite variable) it is effectively the same as saying

```
GPS1.Pos.Lat = {seconds=60 rate_hz=1}
GPS1.Pos.Lon = {seconds=60 rate_hz=1}
```

Unless a variable is explicitly added to the VarHistory section the history will not be available. The amount of data recorded to a variables history is defined by the parameters specified for the variable. You should not attempt to add

history that covers an excessive number of samples or over a very long period of time as this can affect the performance of the application.

If the rate_hz is omitted then the history will be added to on each update of the given variable.

For additional information see section A.9 History Objects

A.3.1.15 [LogFile1]…

The LogFileN sections set up individual log files

Example:

```
[LogFile1]
Title                                = GPS
Type                                 =
Standard
BaseFileName                         =
GpsTest_
DurationInHours                      = 0.2
RateInSeconds                        = 4
MaxFileSizeInBytes = 1600000
Field1 = System.Date          {heading =
"System Date"}
Field2 = System.Time          {heading =
"System Time"}
Field3 = GPS1.Pos.Lat         {heading =
"Lat" }
Field4 = GPS1.Pos.Lon         {heading =
"Lon" }
```

Another example:

```
[LogFile3]
Title                       = GyroData
Type                        = Message
BaseFileName                = GyroData_
DurationInHours             = 0.2
MaxFileSizeInBytes          = 1200000
Trigger                  = GYRO1
Field1 = System.Date {heading="System Date"}
Field2 = System.Time {heading="System Time"}
Field3 = bin2hex(PortInput.GYRO1) {heading="Gyro Data"}
```

A.3.1.15.1 Title=

This key allows the log file to be given a meaningful name. It is a required field but it has limited uses and is not used for naming the files

A.3.1.15.2 Type=

This key identifies the type of log file must be set to a one of the following values

- Normal
  Normal logs are all written at the same interval. The interval for normal logs is entered in the logging configuration dialog. You can specify the interval as a time and/or distance moved by SP1/2 or 3. All normal logs are written simultaneously. Writes to standard logs are also triggered by fixing (or NavSlack events) and when this happens the event log(s) will also be written at the same time. Whenever the normal logs are written to the variable Logging.FixNumber is incremented. For writes triggered by a NavSlack event the Logging.EventNumber is also incremented. You define the fields of a normal log in the {LogFileN] section. You cannot specify individual rates (or triggers) for normal logs (RateInSeconds does not apply)

- Standard
  Standard logs are written at a periodic interval such as once a second. You define the fields of a standard log in the {LogFileN] section. Each standard logs can be written at a different rate. Logging.FixNumber is NOT incremented when writing to standard logs.

- Message
  Message logs are typically written when triggered to write by the arrival of an incoming message. The fields are defined in the same way as for standard logs.

- Output
  Output logs are like custom outputs and can therefore also be written at a regular rate or triggered by incoming data. However the data written to an output log has no column headings and does not have to be in CSV format. An output log can also have multiple messages just like a custom output. For an output log the fields making up the data are defined in custom output format(s) and not in the log file definition itself. Logging rates and triggers are ignored and the ones defined for the custom output are used instead.

- Alert
  Alert logs are exactly the same as the built in alert logs except they are recorded to your chosen location and subject to your own size and time limits. You do have to configure the fields for an alert log and you would typically add the variables System.Timestamp and Alert.Description but you may add additional fields if so desired. Alert logs are written to whenever the system records a new alert. You cannot specify a RateInSeconds or a trigger condition.

- Anchor
  Anchor logs are written to only as a result of anchor operations and are designed for recording deployment positions and recovery positions of each anchor. You have to configure the variables that you wish to log. You cannot specify a RateInSeconds or a trigger condition.

- HPR
  HPR logs are designed to record HPR beacon positions. A HPR log is written to once for each beacon position received. A message or output log can be configured to perform a similar function but can only record the raw data. A HPR log can record the decoded HPR data. You cannot specify a RateInSeconds or a trigger condition.

A.3.1.15.3 BaseFileName=

The BaseFileName key specifies the prefix on the name of the log file.

Log files are named by combining the prefix here with the date and time when the log file is started.

A.3.1.15.4 Extension=

The extension key specifies a file name extension. The default is CSV. If you are logging raw binary data using a Type=Output log file then you may wish to specify a different file extension.

A.3.1.15.5 DurationInHours=

This keys value sets the maximum duration of the log file in hours. When this time period expires a new log file will be started and the previous one will be copied to the remote destination.

A.3.1.15.6 RateInSeconds=

This keys value sets the recording rate for the log file. Fractions of a second are allowed. A rate of zero is not allowed. Setting the rate at close to zero will mean that logging will be performed as fast as possible but if set to fast then the required rate might not be achievable and warnings may occur.

The RateInSeconds does not need to be specified as logging can be triggered to occur on arrival of incoming messages

using . However you must specify either a rate in seconds or a trigger using Trigger.

You can specify both a rate in seconds and a trigger condition but this will mean that logging will occur at the specified rate AND when the trigger condition is true.

For Type=Output log files the trigger and rates are ignored. The log file is written to as a result of the output being triggered.

RateInSeconds can only be specified for Standard logs.

(Normal logs are all written at the rates defined in the logging configuration dialog.)

A.3.1.15.7 MaxFileSizeInBytes=

This keys value places a constraint on the file size. When the file is about to exceed the specified size a new log file will be started and the previous one copied to the remote destination.

If DurationInHours (or MaxLines) are also specified then it is the first of these conditions that will cause a new log file to be created.

A.3.1.15.8 MaxLines=

To limit the maximum number of lines in a file you can specify this limit here.

A.3.1.15.9 Trigger=

This keys value sets a number of possible trigger conditions that will cause a write to the log file to occur. You can use any variable name or a list of variable names as the value of this key.

If you specify more than one variable name then logging will occur as each of the variables change. In most cases a single variable should be specified.

Trigger can only be specified for **Standard** logs.

For **Type=Output** log files the trigger and rates are ignored. The log file is written to as a result of the output being triggered. You must instead specify the trigger condition for the output [CustomOutputFormat1] etc. rather than the log file itself.

For **Type=Normal** the rate is fixed and writes only occur at the normal logging rate. Neither **Trigger** nor **RateInSeconds** can be specified. See A.3.1.15.2 Type= (p.199)

A.3.2 How to Decode Fields

The INI file field decoder works in the same way for almost all the sections. The variable is parsed using 3 numbers.

Field = a,b,c where Field will hold the extracted data.

“a” is the field count number (where fields are delimited).
“b” is the first character within a field for the data extract.
“c” is the number of characters to be extracted.

If (b) and (c) are both zero then (a) is used as a field number and the position is calculated by counting delimiters.

For further detailed information see A.3.1.3.6 Field specifiers (p.169)

A.4 Machine.acl.INI

This file contains the address and permissions of computers that are allowed to

- See latitude and longitude positions on the screen.
- Modify geodetic settings.
- Record anchor handling operations.

For all of these options, 0 = Disabled, 1 = Enabled. These options work in conjunction with the BSPEngine.INI file [System] section

Example Machine.acl.INI

```
[OPS1]
AllowPositionDisplay = 1
AllowCoordSysChanges = 1
AllowAnchorHandling = 1

[Testroom]
AllowPositionDisplay = 0
AllowCoordSysChanges = 0
```

A.5 Communications Device Names

Device Name	*Purpose*
GPS1	Primary GPS
GPS2	Additional GPS
GPS3	Additional GPS
GYRO1	Primary Gyro
GYRO2	Additional Gyro
GYRO3	Additional Gyro
ECHO1	Primary echo sounder
ECHO2	Additional echo sounder
ECHO3	Additional echo sounder
RP01	Primary motion sensor
RP02	Additional motion sensor
RP03	Additional motion sensor

Device Name	*Purpose*
OP_01	General purpose outputs
...	...
OP_16	...
IP_01	General purpose inputs
...	...
IP_32	...
RTT_01	Special positioning inputs
...	...
RTT_08	...
AIS_01	AIS input

For each communications device name there is a variable called PortInput.DeviceName where DeviceName is replaced with the name from the above table. These variables hold the last received data on the corresponding port. If you wish

to log raw data directly from these variables or turnaround an incoming stream on an input port you should trigger this on update of the PortInput.DeviceName variable.

For PortInput.AIS_01 there is a related variable called PortInput.AIS_01.Filtered which contains the resulting filtered AIS data if filtering has been enabled. If filtering has not been configured then the value of this variable is the unfiltered AIS data. For more information see A.12 AIS Filtering

A.6 Built in Variables

This section describes the built in variables that can be used for logging or in Javascript expressions to create additional variables.

Note all variables with a number suffix in the name can also be accessed as arrays in Javascript. Arrays use a 0 based index but the number suffixes are 1 based.

GPS1.Pos.Lat

is the same variable as

GPS[0].Pos.Lat

The array notation is more useful when you want to use another variable as the array index.

Array aliases are also created for your own user defined variables if you apply a number suffix when defining the variable.

Do not use a number suffix in variable names if this is not what was intended.

Note a variable name such as User.TSS340 will create an array and define element 339. If you want to include a number in a variable name try to avoid putting it at the end of the name like this.

A.6.1 Variable Names

Quick reference summary of built in variables

Variable Name	Description
AHT.Act.Time	Time (time) at which the action was requested.
AHT.Act.Date	Time (date) at which the action was requested.
AHT.Action	Anchor action requested.
AHT.Anchor.Name	Name of the anchor for which an action is requested.
AHT.Anchor.Owner	Name of the barge owning the anchor.
AHT.Drop.DeltaM.X	True X difference in position between the intended and actual drop position in metres.
AHT.Drop.DeltaM.Y	True Y difference in position between the intended and actual drop position in metres.
AHT.Drop.GDelta.X	X difference in grid position between the intended and actual drop position.
AHT.Drop.GDelta.Y	The Y difference in grid position between the intended and actual drop position.
AHT.Drop.Grid.Easting	Actual grid position (easting) at which the anchor was dropped.
AHT.Drop.Grid.Northing	Actual grid position (northing) at which the anchor was dropped.
AHT.Drop.Pos.Lat	Actual position (latitude) at which the anchor was dropped.

Variable Name	Description
AHT.Drop.Pos.Lon	Actual position (longitude) at which the anchor was dropped.
AHT.Grid.Easting	Last grid position (easting) at which an anchor was either dropped or recovered.
AHT.Grid.Northing	Last grid position (northing) at which an anchor was either dropped or recovered.
AHT.Pickup.DeltaM.X	True X difference in position between the the drop position and recovery position in metres.
AHT.Pickup.DeltaM.Y	True Y difference in position between the the drop position and recovery position in metres.
AHT.Pickup.GDelta.X	X difference in grid position between the drop position and recovery position.
AHT.Pickup.GDelta.Y	Y difference in grid position between the drop position and recovery position.
AHT.Pickup.Grid.Easting	Actual grid position (easting) from which the anchor was recovered.
AHT.Pickup.Grid.Northing	Actual grid position (northing) from which the anchor was recovered.
AHT.Pickup.Pos.Lat	Actual position (latitude) from which the anchor was recovered.
AHT.Pickup.Pos.Lon	Actual position (longitude) from which the anchor was recovered.
AHT.Pos.Lat	Last position (latitude) at which an anchor was either dropped or recovered.
AHT.Pos.Lon	Last position (longitude) at which an anchor was either dropped or recovered.
AHT.Requestor	Name of the barge or tug requesting an anchor action.
AHT.Target.Grid.Easting	Target drop grid position (easting) for the anchor deployment.
AHT.Target.Grid.Northing	Target drop grid position (northing) for the anchor deployment.

Variable Name	Description
AHT.Target.Pos.Lat	Target drop position (latitude) for the anchor deployment.
AHT.Target.Pos.Lon	Target drop position (longitude) for the anchor deployment.
AHT.Tug.Name	Name of the tug being referred to in the anchor request.
Alert.Description	Last recorded system alert string.
AutoPilot.Direction	Direction 'L' or 'R' that an autopilot should steer in order to stay on the active route line.
AutoPilot.ReversedDirection	Direction 'L' or 'R' that an autopilot should steer in order to stay on the active route line if the vessel needs to move in reverse.
AutoPilot.SP2.Direction	Direction 'L' or 'R' that an autopilot (steering SP2) should steer in order to stay on the active route line.
AutoPilot.SP2.ReversedDirection	Direction 'L' or 'R' that an autopilot (steering SP2) should steer in order to stay on the active route line.
Beacon.ID	ID of the beacon from which a position was last received.
Beacon.Pos.Lat	Latitude value received from the last HPR or RTT message.
Beacon.Pos.Lon	Longitude value received from the last HPR or RTT message.
Beacon.X	Raw X value received from the last HPR or RTT message.
Beacon.Y	Raw Y value received from the last HPR or RTT message.
Beacon.Z	Raw Z value received from the last HPR or RTT message.
Cable.AUX1.Length	Cable length in kilometres for the AUX2 cable channel (cable operations).

Variable Name	Description
Cable.AUX1.SlackFromSectionStart	Smoothed slack percentage from section start for AUX1 cable channel (cable operations).
Cable.AUX1.Smoothed.Slack	For cable operations.
Cable.AUX1.Smoothed.Speed	Cable speed in km/h for the AUX1 cable channel (cable operations).
Cable.AUX1.Smoothed.Tension	Cable tension in (kN) for the AUX1 cable channel (cable operations).
Cable.AUX1.Speed	Raw cable speed in km/h for the aux1 cable channel (cable operations).
Cable.AUX1.Tension	Raw cable tension in (kN) for the AUX1 cable channel (cable operations).
Cable.AUX2.Length	Cable length in kilometres for the AUX2 cable channel (cable operations).
Cable.AUX2.SlackFromSectionStart	Smoothed slack percentage from section start for AUX2 cable channel (cable operations).
Cable.AUX2.Smoothed.Slack	Smoothed slack percentage for AUX2 cable channel (cable operations).
Cable.AUX2.Smoothed.Speed	Cable speed in km/h for the AUX2 cable channel (cable operations).
Cable.AUX2.Smoothed.Tension	Cable tension in (kN) for the AUX2 cable channel (cable operations).
Cable.AUX2.Speed	Raw cable speed in km/h for the aux2 cable channel (cable operations).
Cable.AUX2.Tension	Raw cable tension in (kN) for the AUX2 cable channel (cable operations).
Cable.ControlSpeed	Speed in m/s being automatically demanded on systems with automatic cable control.
Cable.DistanceDeviation	Distance deviation is based on slack and cable speed.
Cable.Engine1.CableOut	Cable out length from cable engine #1

Variable Name	Description
Cable.Engine1.Tension	Cable tension from cable engine #1
Cable.Engine2.CableOut	Cable out length from cable engine #2
Cable.Engine2.Tension	Cable tension from cable engine #2
Cable.Engine3.CableOut	Cable out length from cable engine #3
Cable.Engine3.Tension	Cable tension from cable engine #3
Cable.Engine4.CableOut	Cable out length from cable engine #4
Cable.Engine4.Tension	Cable tension from cable engine #4
Cable.Factory.Length	Factory length of the cable (cable operations).
Cable.Grid.Easting	Grid position of (cable detector) (easting).
Cable.Grid.Northing	The grid position of (cable detector) (northing).
Cable.PLC1.Raw.Count	Encoder counter value from cable engine #1 this value will be scaled by the navslack encoder/strain gauges scaling factors.
Cable.PLC1.Raw.Tension	Raw load cell tension value from cable engine #1.
Cable.PLC2.Raw.Count	Encoder counter value from cable engine #2 this value will be scaled by the navslack encoder/strain gauges scaling factors.
Cable.PLC2.Raw.Tension	Raw load cell tension value from cable engine #2.
Cable.PLC3.Raw.Count	Encoder counter value from cable engine #3 this value will be scaled by the navslack encoder/strain gauges scaling factors
Cable.PLC3.Raw.Tension	Raw load cell tension value from cable engine #1.
Cable.Pos.Alt	Geodetic position of (cable detector) (altitude).
Cable.Pos.Lat	Geodetic position of (cable detector) (longitude).
Cable.Pos.Lon	Geodetic position of (cable detector) (latitude).
Cable.Primary.Length	Cable length in kilometres for the primary cable channel (cable operations).

Variable Name	Description
Cable.Primary.SlackFromSectionStart	Smoothed slack percentage from section start for primary cable channel (cable operations).
Cable.Primary.Smoothed.Slack	Smoothed slack percentage for primary cable channel (cable operations).
Cable.Primary.Smoothed.Speed	Smoothed cable speed in km/h for the primary cable channel (cable operations).
Cable.Primary.Smoothed.Tension	Smoothed cable tension in (kN) for the primary cable channel (cable operations).
Cable.Primary.Speed	Raw cable speed in km/h for the primary cable channel (cable operations).
Cable.Primary.Tension	Raw cable tension in (kN) for the primary cable channel (cable operations).
Cable.RouteDistance	Distance along cable route (cable operations).
Cable.TargetSlack	Target slack percentage (cable operations)
Cable.TargetSpeedKmh	Desired cable speed based on required slack (see Cable.TargetSlack).
Cable.TargetTension	Target cable tension (cable operations)
Clara.AutoSolveMode	Clara calculation auto solve mode 0 =no autosolve, 1=use inclinometer angle, 2=use top tension measurement.
Clara.CableInfo	Clara calculation cable name details.
Clara.MBTension	Clara calculation manually set bottom tension.
Clara.MSeabedSlope	Clara calculation manual seabed slope (entered by user).
Clara.UserAdjust	Clara calculation adjustment verb (for internal use).
Clara.UseRouteDepth	If the Clara calculation uses route slope then this has the value 1.
Clara.UseRouteSlope	If the Clara calculation uses route slope then this has the value 1.

Variable Name	Description
GPS1.Altitude	Altitude reported by the GPS receiver #1. ***(Equivalent variables are available for GPS2 and 3)***
GPS1.AltitudeWGS84	Altitude reported by the GPS receiver #1.! BSPEngine expects all GPS receivers to output position in WGS84 and this means the altitude is also with respect to the WGS84 geoid.The altitude here is the altitude of the antenna not the CRP.
GPS1.CRP.DX	Computed DX value of CRP position computed by GPS1.
GPS1.CRP.DY	Computed DY value of CRP position computed by GPS1.
GPS1.CRP.DZ	Computed DZ value of CRP position computed by GPS1.
GPS1.CRP.Pos.Alt	Computed altitude value of CRP position derived from GPS1 in your working datum/vertical reference
GPS1.CRP.Pos.Lat	Computed CRP position (latitude) derived from GPS1 in your working datum
GPS1.CRP.Pos.Lon	Computed CRP position (longitude) derived from GPS1 in your working datum
GPS1.CRP.WGS84.Pos.Alt	Computed altitude value of CRP derived from GPS1 in WGS84
GPS1.CRP.WGS84.Pos.Lat	Computed CRP position (latitude) derived from GPS1 in WGS84
GPS1.CRP.WGS84.Pos.Lon	Computed CRP position (longitude) derived from GPS1 in WGS84
GPS1.Date	Time (date) of GPS1 (DD/MM/YYYY) as received from GPS receiver #1.
GPS1.DatumShifted.Pos.Alt	Position of GPS1 (altitude).
GPS1.DatumShifted.Pos.Lat	Geodetic position of GPS1 (latitude).
GPS1.DatumShifted.Pos.Lon	Geodetic position of GPS1 (longitude).

Variable Name	Description
GPS1.GeoidalSeparation	Geoidal separation reported by the GPS receiver #1.
GPS1.Grid.Easting	Grid position of GPS1 (easting).
GPS1.Grid.Northing	Grid position of GPS1 (northing).
GPS1.HDOP	HDOP (horizontal dilution of precision) as received from the GPS receiver #1.
GPS1.PDOP	PDOP (dilution of precision) as received from the GPS receiver #1.
GPS1.Pos.Lat	Geodetic position reported by GPS1 (latitude).
GPS1.Pos.Lon	Geodetic position reported by GPS1 (longitude).
GPS1.Quality	Quality indicator reported by GPS receiver #1.
GPS1.Sats	Number of satellites/ground stations in view to GPS receiver #1
GPS1.Time	Time (time) of GPS1 (HH:MM:SS.SS) as received from GPS receiver #1.
GPS1.VDOP	VDOP (vertical dilution of precision) as received from the GPS receiver #1.
GPS1.GPS2.Heading	Computed raw vessel heading derived from GPS1 to GPS2 vector.
GPS2.GPS3.Heading	Computed raw vessel heading derived from GPS2 to GPS3 vector.
GPS3.GPS1.Heading	Computed raw vessel heading derived from GPS3 to GPS1 vector.
Gyro1.Corr.Heading	Adjusted heading in degrees reported by GYRO #1 ***(Gyro variables are also available for Gyro2 and 3.)***
Gyro1.Heading	Adjusted heading in degrees reported by GYRO #1
Gyro1.Message	Message string from the gyro #1 input.
Gyro1.Raw.Heading	Raw heading in degrees reported by GYRO #1

Variable Name	Description
HPR.Ancilliary.Heading	Heading value from HPR message (only a few HPR message types are likely to have this data)
HPR.Ancilliary.Heave	Heave value from HPR message (only a few HPR message types are likely to have this data)
HPR.Ancilliary.Pitch	Pitch value from HPR message (only a few HPR message types are likely to have this data)
HPR.Ancilliary.Roll	Roll value from HPR message (only a few HPR message types are likely to have this data)
Logging.Backup1.AnticipatedSize	***(These variables are available for each backup log file.)*** Anticipated size of the backup log file #1 in bytes
Logging.Backup1.FileSize	Size of the backup log file #1 in bytes
Logging.Backup1.Unc	UNC filename of the backup log file #1
Logging.Cable.Line.Name	Cable line name DEPRECATED (do not use) will only output empty string.
Logging.Cable.Line.No	Cable line number (string).
Logging.Cable.Type	Cable type (string).
Logging.Comment	Fix comment string.
Logging.Config1.LogType	***These variables are available for each configured log file*** Configured type of the primary log file #1
Logging.Config1.Name	Caption name of log file #1
Logging.Description	Fix description string.
Logging.EventNo	Last event number used when writing to CSV log files.
Logging.FixedSP	Last fix SP when writing to CSV log files.
Logging.FixNo	Last fix number used when writing to CSV log files.
Logging.Primary1.AnticipatedSize	***These variables are available for each primary log file*** Anticipated size of the primary log file #1 in bytes

Variable Name	Description
Logging.Primary1.FileSize	Size of the primary log file #1 in bytes
Logging.Primary1.Unc	UNC filename of the primary log file #1
MRU1.Heave	**These variables are available for MRU1 to MRU3** The heave reported by motion sensor #1 device (in metres).
MRU1.Pitch	Pitch reported by motion sensor #1 device (in degrees).
MRU1.Roll	Roll reported by motion sensor #1 device (in degrees).
Option.SpeedGaugeKmh.Max	Speed gauge range max This variable gives the minimum range of the speed gauge in Navslack
Option.SpeedGaugeKmh.Min	Speed gauge range min This variable gives the minimum range of the speed gauge in Navslack
Option.TensionGaugeKN.Max	Tension gauge range max This variable gives the minimum range of the tension gauge in Navslack
Option.TensionGaugeKN.Min	Tension gauge range min This variable gives the minimum range of the tension gauge in Navslack
PrimaryGPS.Altitude	Altitude reported by the primary GPS receiver.
PrimaryGPS.AltitudeWGS84	Altitude reported by the primary GPS receiver.
PrimaryGPS.GeoidalSeparation	Altitude reported by the primary GPS receiver.
PrimaryGPS.HDOP	Primary GPS HDOP (horizontal dilution of precision) as received from the primary GPS receiver.
PrimaryGPS.Quality	Quality indicator reported by the primary GPS receiver.
PrimaryGPS.Sats	Number of satellites/ground stations in view to the GPS receiver.

Variable Name	Description
Remotes.Vessel1.ID	***These variables are available for each remote vessel.*** ID of remote vessel #1
Remotes.Vessel1.Name	Name of remote vessel #1
Remotes.Vessel1.Push.Avg.Latency	Averaged amount of time in seconds that it is taking BSPEngine to push status information to remote vessel #1
Remotes.Vessel1.Push.Latency	Amount of time in seconds that it last took BSPEngine to push status information to remote vessel #1
Remotes.Vessel1.SP1.Offset.Name	For remote vessel #1, name of the vessel offset which is currently SP1
Remotes.Vessel1.SP1.Pos.Alt	For remote vessel #1, The altitude of the SP1 offset.
Remotes.Vessel1.SP1.Pos.Lat	For remote vessel #1, The latitude of the SP1 offset.
Remotes.Vessel1.SP1.Pos.Lon	For remote vessel #1, The longitude of the SP1 offset.
Remotes.Vessel1.Stats.Avg.TimeDelta	Measured average time difference in seconds between local and remote vessels system clocks.
Remotes.Vessel1.System.Timestamp	System time and date string from remote vessel #1
Remotes.Vessel1.Time.Delta	Measured time difference in seconds between local and remote vessels system clocks.
Remotes.Vessel1.Time.Estimated.Timestamp	System time and date string from remote vessel #1 taking account of push latency.
Route.Direction	Direction of the currently active route.
Route.Name	Name of the active route line
Route.Target.Name	Name of the current target.

Variable Name	Description
Route.Target1.Pos.Lat	***These variables are available for the auxiliary targets Target1 to Target4*** Latitude of the current target #1
Route.Target1.Pos.Lon	Latitude of the current target #1
Route.Target1.WGS84.Pos.Lat	Latitude of the current target #1
Route.Target1.WGS84.Pos.Lon	Latitude of the current target #1
RTT_01.Altitude	***These variables are available for each RTT channel RTT_01 to RTT_08*** Altitude reported by the RTT_01 input.
RTT_01.Heading	Raw heading in degrees reported by RTT #1
RTT_01.Pos.Lat	Geodetic position of RTT_01 (latitude).
RTT_01.Pos.Lon	Geodetic position of RTT_01 (longitude).
RTT_01.WaterDepth	Water depth reported by the RTT_01 input.
Ship.AvgWaterLine	Averaged altitude of the ship waterline with respect to the working datum.
Ship.AvgWaterLineWGS84	Averaged altitude of the ship waterline with respect to the WGS84 geoid.
Ship.CableEngines.PrimaryChannel	Primary cable engine channel number (cable operations)
Ship.CRP.AltitudeWGS84	Altitude reported by the primary GPS receiver but translated to the CRP position.
Ship.DesiredSpeedKmh	Desired ship speed in kilometres per hour.
Ship.Draft	Distance from the keel to the waterline.
Ship.EchoSounderDepth	Raw water depth reading obtained from the primary echo sounder.
Ship.EchoSounderDepth1	***Variables also available for echo sounder 2 and 3*** Raw water depth reading obtained from echo sounder #1.

Variable Name	Description
Ship.GeoidWaterDepth	This is water depth relative to the current vertical datum (from primary depth sounder)
Ship.GeoidWaterDepth1	***Variables also available for echo sounder 2 and 3*** This is water depth relative to the current vertical datum (from depth sounder #1)
Ship.GPS.AltitudeWGS84	Altitude reported by the primary GPS receiver.
Ship.GPS.GeoidalSeparation	Geoidal separation reported by the primary GPS receiver.
Ship.GPS.HDOP	Primary GPS HDOP (horizontal dilution of precision) as received from the primary GPS receiver.
Ship.GPS.Pos.Alt	The altitude reported by the primary GPS receiver.
Ship.GPS.Pos.Lat	Latitude reported by the primary (or integrated) GPS receiver.
Ship.GPS.Pos.Lon	Longitude reported by the primary (or integrated) GPS receiver.
Ship.GPS.Quality	Quality indicator reported by the primary GPS receiver.
Ship.GPS.ReceiverFlags	A set of bitflags describing how the primary GPS system is configured.
Ship.GPS.Sats	Number of satellites/ground stations in view to the main GPS receiver.
Ship.GPS.VTG.Course	Course as reported by the VTG message (if present) from the primary GPS.
Ship.GPS.VTG.Speed	Ship speed as reported by the GPS VTG message (if present) from the primary GPS.
Ship.GridHeading	Grid heading based on delta easting/northing and not a true direction.
Ship.Gyro.ReceiverFlags	A set of bitflags describing how the primary Gyro system is configured.
Ship.Heading	Vessel heading from the primary gyro (possibly adjusted with a fixed calibration offset).

Variable Name	Description
Ship.Kalman.CMG	Smoothed ship speed in kilometres per hour (calculated using Kalman filter).
Ship.Kalman.Pos.Lat	Smoothed postion of the ship as calculated by the Kalman filter.
Ship.Kalman.Pos.Lon	Smoothed postion of the ship as calculated by the Kalman filter.
Ship.Kalman.Speed	Smoothed ship speed in kilometres per hour (calculated using Kalman filter).
Ship.KeelHeight	Distance from the keel of the ship from the CRP.
Ship.LaybackPoint	The name of the vessel offset which is currently the default layback offset for all mobiles
Ship.Motion.Heading	Heading of the vessel (in degrees) as obtained from a motion sensor.
Ship.Motion.Heave	Heave of the vessel (in metres) as obtained from a motion sensor.
Ship.Motion.Pitch	Pitch of the vessel (in degrees) as obtained from a motion sensor.
Ship.Motion.Roll	Roll of the vessel (in degrees) as obtained from a motion sensor.
Ship.MRU.ReceiverFlags	A set of bitflags describing how the primary MRU system is configured.
Ship.Offsets.Grid1.Easting	***These variables are available for the first 16 vessel offsets*** Position (easting) of vessel offset #1.
Ship.Offsets.Grid1.Northing	Position (northing) of vessel offset #1.
Ship.Offsets.Pos1.Alt	Position (altitude) of vessel offset #1.
Ship.Offsets.Pos1.Elev	Elevation from seabed to vessel offset.
Ship.Offsets.Pos1.Lat	Position (latitude) of vessel offset #1.
Ship.Offsets.Pos1.Lon	Position (longitude) of vessel offset #1.

Variable Name	Description
Ship.Offsets.WGS84.Pos1.Alt	Position (altitude) of vessel offset #1 converted to WGS84.
Ship.Offsets.WGS84.Pos1.Lat	Position (latitude) of vessel offset #1 converted to WGS84.
Ship.Offsets.WGS84.Pos1.Lon	Position (longitude) of vessel offset #1 converted to WGS84.
Ship.PrimaryGyro.Message	Message string from the primary gyro input.
Ship.RawSpeedKmh	Raw ship speed in kilometres per hour.
Ship.SP1.Grid.Easting	Grid position of SP1 (easting).
Ship.SP1.Grid.Northing	Grid position of SP1 (northing).
Ship.SP1.Pos.Alt	Altitude position of SP1.
Ship.SP1.Pos.Elev	Elevation position of SP1 from the seabed.
Ship.SP1.Pos.Lat	Geodetic position of SP1 (latitude).
Ship.SP1.Pos.Lon	Geodetic position of SP1 (longitude).The position given here is in your selected working datum which is not necessarily WGS84.
Ship.SP1.Route.Arc.DOL	The route arc DOL value of SP1.
Ship.SP1.Route.Arc.KP	Route arc KP value of SP1.
Ship.SP1.Route.DOL	Route DOL value of SP1.
Ship.SP1.Route.Grid.DOL	Route GRID DOL value of SP1.
Ship.SP1.Route.Grid.KP	Route GRID KP value of SP1.
Ship.SP1.Route.KP	Route KP value of SP1.
Ship.Speed	Smoothed ship speed in kilometres per hour.
Ship.SpeedKmh	Ship speed in kilometres per hour.
Ship.SpeedMS	Smoothed ship speed in metres per second.
Ship.VDatumShift	Vertical datum shift to convert from the WGS84 geoid altitude to your selected vertical reference.

Variable Name	Description
Ship.WaterDepth	True water depth from the waterline.
Ship.WaterDepth1	**Also available for depth sounder 2 and 3** True water depth from the waterline (from depth sounder #1) This is water depth from the echo sounder adjusted to CRP level then compensated for draft and heave
Ship.WaterLine	Altitude of the ship waterline with respect to the working datum / vertical reference geoid.
Ship.WaterLineWGS84	Altitude of the ship waterline with respect to the WGS84 geoid.
SP1.Averaged.CMG	Time averaged SP1 course made good in degrees.
SP1.Averaged.Speed	Time averaged SP1 speed in metres per second.
SP1.Averaged.SpeedKmh	Time averaged SP1 speed in kilometres per hour.
SP1.Date	Time (date) of SP1/GPS (DD/MM/YYYY) as received from the primary GPS receiver.
SP1.GPS.AltitudeWGS84	Altitude reported by the primary GPS receiver.
SP1.GPS.GeoidalSeparation	Altitude reported by the primary GPS receiver.
SP1.GPS.HDOP	Primary GPS HDOP (horizontal dilution of precision) as received from the primary GPS receiver.
SP1.GPS.PDOP	Primary GPS PDOP (primary dilution of precision) as received from the primary GPS receiver.
SP1.GPS.Quality	Quality indicator reported by the primary GPS receiver.
SP1.GPS.Sats	Number of satellites/ground stations in view to the GPS receiver.
SP1.GPS.VDOP	Primary GPS PDOP (primary dilution of precision) as received from the primary GPS receiver.
SP1.Grid.Easting	Grid position of SP1 (easting).
SP1.Grid.Northing	Grid position of SP1 (northing).

Variable Name	Description
SP1.KP	Route KP value of SP1.
SP1.Offset.Name	Name of the vessel offset which is currently SP1
SP1.Pos.Alt	Altitude position of SP1 (Alias for Ship.SP1.Pos.Alt).
SP1.Pos.Lat	Geodetic position of SP1 (latitude).
SP1.Pos.Lon	Geodetic position of SP1 (longitude).
SP1.Route.DOL	Route DOL value of SP1.
SP1.Route.Grid.DOL	Route GRID DOL value of SP1.This is the perpendicular distance from route line.On the outer corner of an alter course it is the distance from the corner(in which case the KP value will be the KP of the corner).Alter course radii are ignored by this calculation
SP1.Route.Grid.KP	Route GRID KP value of SP1.
SP1.Route.KP	Route KP value of SP1.
SP1.Route.SeabedSlope	Route KP (as surveyed) seabed slope value under SP1.
SP1.Route.Section.Bearing	True bearing of the current route section.
SP1.Route.Target.Bearing	True bearing in degrees from SP1 to the current target.
SP1.Route.Target.Range	Range in metres from SP1 to the current target.
SP1.Route.TerrainDist	Route terrain distance value of SP1.
SP1.Route.WaterDepth	Route KP (as surveyed) water depth value SP1.
SP1.Smoothed.CMG	Smoothed SP1 course made good in degrees.
SP1.Speed	Smoothed SP1 speed in metres per second.
SP1.SpeedKmh	Smoothed SP1 speed in kilometres per hour.
SP1.Target1.Bearing	**These variables are available for auxiliary targets 1 to 4.** True bearing in degrees from SP1 to the auxilliary target #1

Variable Name	Description
SP1.Time	Time of SP1/GPS (HH:MM:SS.SS) as received from the primary GPS receiver.
SP1.WGS84.Pos.Alt	Altitude position of SP1 in WGS84.
SP1.WGS84.Pos.Lat	Geodetic position of SP1 (latitude).
SP1.WGS84.Pos.Lon	Geodetic position of SP1 (longitude).
SP2.Grid.Easting	Grid position of SP2 (easting).
SP2.Grid.Northing	Grid position of SP2 (northing).
SP2.GridHeading	Grid heading based on delta easting/northing and not a true direction.
SP2.Heading	True heading for SP2.
SP2.LaybackBearing	Layback bearing of SP2 from the Ship.LaybackPoint
SP2.LaybackDistance	Layback distance of SP2 from the Ship.LaybackPoint
SP2.LaybackMode	Layback mode of the vehicle which is currently SP2
SP2.Motion.Pitch	Pitch of SP2 (in degrees) as obtained from a motion sensor.
SP2.Motion.Roll	Roll of SP2 (in degrees) as obtained from a motion sensor.
SP2.Name	Name of the vehicle (or vessel offset) which is currently SP2
SP2.Offset.Pos.Name	Name of the vehicle offset which is the positioning offset for SP2 (blank if a vessel offset is used)
SP2.Offset.SP.Name	Name of the vehicle offset which is the steer point offset for SP3 (blank if a vessel offset is used)
SP2.Offsets.Grid1.Easting	***These variables are available for the first 8 mobile offsets.*** Position (easting) of SP2 offset #1.
SP2.Offsets.Grid1.Northing	Position (northing) of SP2 offset #1.

Variable Name	Description
SP2.Offsets.Pos1.Alt	***These variables are available for the first 16 mobile offsets.*** Position (altitude) of SP2 (mobile or ship) offset #1.
SP2.Offsets.Pos1.Elev	Elevation from seabed to mobile/vessel offset.
SP2.Offsets.WGS84.Pos1.Alt	Position (altitude) of SP2 offset #1 converted to WGS84.
SP2.Offsets.WGS84.Pos1.Lat	Position (latitude) of SP2 converted to WGS84 (mobile or ship) offset #1.
SP2.Offsets.WGS84.Pos1.Lon	Position (longitude) of SP2 converted to WGS84 (mobile or ship) offset #1.
SP2.Pos.Alt	Position of SP2 (altitude).
SP2.Pos.Elev	Elevation of SP2 above the seabed
SP2.Pos.Lat	Geodetic position of SP2 (latitude).
SP2.Pos.Lon	Geodetic position of SP2 (longitude).
SP2.Positioning	Positioning mode of the vehicle which is currently SP2
SP2.Relative.DX	Delta X offset of SP2 from the vessel CRP
SP2.Relative.DY	Delta Y offset of SP2 from the vessel CRP
SP2.Relative.DZ	Delta Z offset of SP2 from the vessel CRP
SP2.Route.Arc.DOL	Route arc DOL value of SP2.
SP2.Route.Arc.KP	Route arc KP value of SP2.
SP2.Route.DOL	Route DOL value of SP2.
SP2.Route.Grid.DOL	Route GRID DOL value of SP2.
SP2.Route.Grid.KP	Route GRID KP value of SP2.
SP2.Route.KP	Route KP value of SP2.

Variable Name	Description
SP2.Route.SeabedSlope	Route KP (as surveyed) seabed slope value under SP2.
SP2.Route.Section.Bearing	True bearing of the current route section adjacent to SP2
SP2.Route.TerrainDist	Route terrain distance value of SP2.
SP2.Route.WaterDepth	Route KP (as surveyed) water depth value under SP2.
SP2.Smoothed.CMG	Smoothed SP2 course made good in degrees.
SP2.Smoothed.Speed	Smoothed SP2 speed in metres per second.
SP2.Smoothed.SpeedKmh	Smoothed SP2 speed in kilometres per hour.
SP2.SP1Relative.DX	Delta X offset of SP2 from the vessel SP
SP2.SP1Relative.DY	Delta Y offset of SP2 from the vessel SP
SP2.SP1Relative.DZ	Delta Z offset of SP2 from the vessel SP
SP2.Speed	SP2 speed in kilometres per hour.
SP2.SpeedKmh	SP2 speed in kilometres per hour.
SP2.SpeedMS	SP2 speed in metres per second.
SP2.Target1.Bearing	***These variables are available for the first 8 mobile offsets.*** True bearing in degrees from SP2 to the auxiliary target #1
SP2.Target1.Range	Range in metres from SP2 to the auxiliary target #1
SP2.WaterDepth	Water depth at the SP2 offset position (if SP2 is a mobile)
SP2.WGS84.Pos.Alt	Position of SP2 (altitude) in WGS84.
SP2.WGS84.Pos.Lat	Geodetic position of SP2 (latitude).
SP2.WGS84.Pos.Lon	Geodetic position of SP2 (longitude).
SP3.Grid.Easting	Grid position of SP3 (easting).
SP3.Grid.Northing	Grid position of SP3 (northing).

Variable Name	Description
SP3.GridHeading	Grid heading based on delta easting/northing and not a true direction.
SP3.Heading	True heading for SP3.
SP3.LaybackBearing	Layback bearing of SP3 from the Ship.LaybackPoint
SP3.LaybackDistance	Layback distance of SP3 from the Ship.LaybackPoint
SP3.LaybackMode	Positioning mode of the vehicle which is currently SP3
SP3.Motion.Pitch	Pitch of SP3 (in degrees) as obtained from a motion sensor.
SP3.Motion.Roll	Roll of SP3 (in degrees) as obtained from a motion sensor.
SP3.Name	Name of the vehicle (or vessel offset) which is currently SP3
SP3.Offset.Pos.Name	Name of the vehicle offset which is the positioning offset for SP3 (blank if a vessel offset is used)
SP3.Offset.SP.Name	Name of the vehicle offset which is the steer point offset for SP3 (blank if a vessel offset is used)
SP3.Offsets.Grid1.Easting	***These variables are available for the first 8 offsets*** Position (easting) of SP3 offset #1.
SP3.Offsets.Grid1.Northing	Position (northing) of SP3 offset #1.
SP3.Offsets.Pos1.Alt	***These variables are available for the first 16 offsets*** Position (altitude) of SP3 (mobile or ship) offset #1.
SP3.Offsets.Pos1.Elev	Elevation from seabed to mobile/vessel offset.
SP3.Offsets.Pos1.Lat	Position (latitude) of SP3 (mobile or ship) offset #1.
SP3.Offsets.Pos1.Lon	Position (longitude) of SP3 (mobile or ship) offset #1.

Variable Name	Description
SP3.Offsets.WGS84.Pos1.Alt	Position (altitude) of SP3 offset #1 converted to WGS84.
SP3.Offsets.WGS84.Pos1.Lat	Position (latitude) of SP3 converted to WGS84 (mobile or ship) offset #1.
SP3.Offsets.WGS84.Pos1.Lon	Position (longitude) of SP3 converted to WGS84 (mobile or ship) offset #1.
SP3.Pos.Alt	Position of SP2 (altitude).
SP3.Pos.Elev	Elevation of SP3 above the seabed
SP3.Pos.Lat	Geodetic position of SP3 (latitude).
SP3.Pos.Lon	Geodetic position of SP3 (longitude).
SP3.Positioning	Positioning mode of the vehicle which is currently SP3
SP3.Relative.DX	Delta X offset of SP3 from the vessel CRP
SP3.Relative.DY	Delta Y offset of SP3 from the vessel CRP
SP3.Relative.DZ	Delta Z offset of SP3 from the vessel CRP
SP3.Route.Arc.DOL	Route arc DOL value of SP2.
SP3.Route.Arc.KP	Route arc KP value of SP3.This is the straight line distance along the route.
SP3.Route.DOL	Route DOL value of SP3.
SP3.Route.Grid.DOL	Route GRID DOL value of SP2.
SP3.Route.Grid.KP	Route GRID KP value of SP3.
SP3.Route.KP	Route KP value of SP3.This is the straight line distance along the route.
SP3.Route.SeabedSlope	Route KP (as surveyed) seabed slope value under SP3.
SP3.Route.Section.Bearing	True bearing of the current route section adjacent to SP3
SP3.Route.TerrainDist	Route terrain distance value of SP3.

Variable Name	Description
SP3.Route.WaterDepth	Route KP (as surveyed) water depth value under SP3.
SP3.Smoothed.CMG	Smoothed SP2 course made good in degrees.
SP3.Smoothed.Speed	Smoothed SP3 speed in metres per second.
SP3.Smoothed.SpeedKmh	Smoothed SP3 speed in kilometres per hour.
SP3.SP1Relative.DX	Delta X offset of SP3 from the vessel SP
SP3.SP1Relative.DY	Delta Y offset of SP3 from the vessel SP
SP3.SP1Relative.DZ	Delta Z offset of SP2 from the vessel SP
SP3.Speed	SP3 speed in kilometres per hour.
SP3.SpeedKmh	SP4 speed in kilometres per hour.
SP3.SpeedMS	SP3 speed in metres per second.
SP3.Target1.Bearing	***These variables are available for auxiliary targets Target1 to Target4.*** True bearing in degrees from SP3 to the auxilliary target #1
SP3.Target1.Range	Range in metres from SP3 to the auxilliary target #1
SP3.WaterDepth	Water depth at the SP3 offset position (if SP3 is a mobile)
SP3.WGS84.Pos.Alt	Position of SP3 (altitude) in WGS84
SP3.WGS84.Pos.Lat	Geodetic position of SP3 (latitude).
SP3.WGS84.Pos.Lon	Geodetic position of SP3 (longitude).
System.CommsScannerState	State of the communications accesiblilty scanner thread in BSPEngine.
System.CoordinateSystem	Coordinate system key describing the coordinate setup currently in use.
System.Date	System date in DD/MM/YYYY format.
System.DBR.CablesRevision	Clara database revision.

Variable Name	Description
System.DBR.FixfilesRevision	Overall revision of all project fix xml tree layout files BUT NOT the associated state data files (contents of the !FixLayouts folder but just the .xml files).
System.DBR.FixfullRevision	Overall revision of all project fix xml tree layout files and associated state data files (contents of the !FixLayouts folder).
System.DBR.FixlayoutRevision	Fix layout file revision (!NavFix.cfg).
System.DBR.GeodeticsRevision	Geodetics database revision (!NavGeo.dat).
System.DBR.MobileShapesRevision	Mobile shapes revision.
System.DBR.MobilesRevision	Mobiles database revision.
System.DBR.RoutesRevision	Route database revision.
System.DBR.RoutesShapesRevision	Route shapes revision.
System.DBR.ShipRevision	Vessel definition (database) revision.
System.DBR.VarsRevision	System variable table revision.
System.Time	System time in HH:MM:SS.SS format.
System.Timestamp	System date and time.
System.VMUsage	Number of bytes of virtual memory currently used by BSPEngine.
Target1.Name	***Variables are available for auxiliary targets Target1 to Target4*** Name of the current auxiliary target #1.

A.6.1.1 AHT.Act.Date

[String]

Default caption in log files: "Date"
Attribute flags: TYPE_STRING | TYPE2_DATE | ASSOC_NEXT | SCAN_NOLOG

The time (date) at which the action was requested. This variable is for anchor handling operations and is designed for use with anchor log files. See AHT.Act.Time

A.6.1.2 AHT.Act.Time

[String]

Default caption in log files: "Time"
Attribute flags: TYPE_STRING | TYPE2_TIME | ASSOC_PREV

The time (time) at which the action was requested. This variable is for anchor handling operations and is designed for use with anchor log files.

A.6.1.3 AHT.Action

[String]

Default caption in log files: "Action"
Attribute flags: TYPE_STRING

The anchor action requested. This variable is for anchor handling operations and is designed for use with anchor log files.

A.6.1.4 AHT.Anchor.Name

[String]

Default caption in log files: "Anchor"
Attribute flags: TYPE_STRING

The name of the anchor for which an action is requested. This variable is for anchor handling operations and is designed for use with anchor log files.

A.6.1.5 AHT.Anchor.Owner

[String]

Default caption in log files: "Barge"
Attribute flags: TYPE_STRING

The name of the barge owning the anchor. This variable is for anchor handling operations and is designed for use with anchor log files.

A.6.1.6 AHT.Drop.DeltaM.X

[Real Number]

Default caption in log files: "Delta East (m)"
Default format specifier: "%+.0f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_NOLOG

The true X difference in position between the intended and actual drop position in metres. This variable is for anchor handling operations and is designed for use with anchor log files.

A.6.1.7 AHT.Drop.DeltaM.Y

[Real Number]

Default caption in log files: "Delta North (m)"
Default format specifier: "%+.0f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_NOLOG

The true Y difference in position between the intended and actual drop position in metres. This variable is for anchor handling operations and is designed for use with anchor log files.

A.6.1.8 AHT.Drop.GDelta.X

[Real Number]

Default caption in log files: "Delta East"
Default format specifier: "%+.0f"
Attribute flags: TYPE_DOUBLE | LOG_VALUE | SCAN_NOLOG

The X difference in grid position between the intended and actual drop position.

This is not the same as AHT.Drop.Delta.X which gives the true differnce in metres. Note that grid distances (and directions) are not neccessarily the same as real world distances. This variable is for anchor handling operations and is designed for use with anchor log files.

A.6.1.9 AHT.Drop.GDelta.Y

[Real Number]

Default caption in log files: "Delta North"
Default format specifier: "%+.0f"
Attribute flags: TYPE_DOUBLE | LOG_VALUE | SCAN_NOLOG

The Y difference in grid position between the intended and actual drop position. This is not the same as AHT.Drop.Delta.Y which gives the true differnce in metres. Note that grid distances (and directions) are not neccessarily the same as real world distances. This variable is for anchor handling operations and is designed for use with anchor log files.

A.6.1.10 AHT.Drop.Grid.Easting

[Real Number]

Default caption in log files: "Drop Easting"
Default format specifier: "%+.0f"
Attribute flags: TYPE_DOUBLE | TYPE2_EAST | LOG_VALUE | SCAN_NOLOG

The actual grid position (easting) at which the anchor was dropped. This variable is for anchor handling operations and is designed for use with anchor log files.

A.6.1.11 AHT.Drop.Grid.Northing

[Real Number]

Default caption in log files: "Drop Northing"
Default format specifier: "%+.0f"
Attribute flags: TYPE_DOUBLE | TYPE2_NORTH | LOG_VALUE | SCAN_NOLOG

The actual grid position (northing) at which the anchor was dropped. This variable is for anchor handling operations and is designed for use with anchor log files.

A.6.1.12 AHT.Drop.Pos.Lat

[Real Number]

Default caption in log files: "Drop Latitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LAT | UNIT_ANGLE | LOG_VALUE

The actual position (latitude) at which the anchor was dropped. This variable is for anchor handling operations and is designed for use with anchor log files.

A.6.1.13 AHT.Drop.Pos.Lon

[Real Number]

Default caption in log files: "Drop Longitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LON | UNIT_ANGLE | LOG_VALUE

The actual position (longitude) at which the anchor was dropped. This variable is for anchor handling operations and is designed for use with anchor log files.

A.6.1.14 AHT.Grid.Easting

[Real Number]

Default caption in log files: "Easting"
Default format specifier: "%+.0f"
Attribute flags: TYPE_DOUBLE | TYPE2_EAST | LOG_VALUE | SCAN_NOLOG

The last grid position (easting) at which an anchor was either dropped or recovered. This variable is for anchor handling operations and is designed for use with anchor log files.

A.6.1.15 AHT.Grid.Northing

[Real Number]

Default caption in log files: "Northing"
Default format specifier: "%+.0f"
Attribute flags: TYPE_DOUBLE | TYPE2_NORTH | LOG_VALUE | SCAN_NOLOG

The last grid position (northing) at which an anchor was either dropped or

recovered. This variable is for anchor handling operations and is designed for use with anchor log files.

A.6.1.16 AHT.Pickup.DeltaM.X

[Real Number]

Default caption in log files: "Drag East (m)"
Default format specifier: "%+.0f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_NOLOG

The true X difference in position between the the drop position and recovery position in metres. This gives an indication of anchor drag amount. This variable is for anchor handling operations and is designed for use with anchor log files.

A.6.1.17 AHT.Pickup.DeltaM.Y

[Real Number]

Default caption in log files: "Drag North (m)"
Default format specifier: "%+.0f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_NOLOG

The true Y difference in position between the the drop position and recovery position in metres. This gives an indication of anchor drag amount. This variable is for anchor handling operations and is designed for use with anchor log files.

A.6.1.18 AHT.Pickup.GDelta.X

[Real Number]

Default caption in log files: "Drag East"
Default format specifier: "%+.0f"
Attribute flags: TYPE_DOUBLE | LOG_VALUE | SCAN_NOLOG

The X difference in grid position between the drop position and recovery position. This is not the same as AHT.Pickup.Delta.X which gives the true differnce in metres. This gives an indication of anchor drag amount. Note that grid distances (and directions) are not neccessarily the same as real world distances. This variable is for anchor handling operations and is designed for use with anchor log files.

A.6.1.19 AHT.Pickup.GDelta.Y

[Real Number]

Default caption in log files: "Drag North"
Default format specifier: "%+.0f"
Attribute flags: TYPE_DOUBLE | LOG_VALUE | SCAN_NOLOG

The Y difference in grid position between the drop position and recovery position. This is not the same as AHT.Pickup.Delta.X which gives the true differnce in metres. This gives an indication of anchor drag amount. Note that grid distances (and directions) are not neccessarily the same as real world distances. This variable is for anchor handling operations and is designed for use with anchor log files.

A.6.1.20 AHT.Pickup.Grid.Easting

[Real Number]

Default caption in log files: "Pickup Easting"
Default format specifier: "%+.0f"
Attribute flags: TYPE_DOUBLE | TYPE2_EAST | LOG_VALUE | SCAN_NOLOG

The actual grid position (easting) from which the anchor was recovered. This variable is for anchor handling operations and is designed for use with anchor log files.

A.6.1.21 AHT.Pickup.Grid.Northing

[Real Number]

Default caption in log files: "Pickup Northing"
Default format specifier: "%+.0f"
Attribute flags: TYPE_DOUBLE | TYPE2_NORTH | LOG_VALUE | SCAN_NOLOG

The actual grid position (northing) from which the anchor was recovered. This variable is for anchor handling operations and is designed for use with anchor log files.

A.6.1.22 AHT.Pickup.Pos.Lat

[Real Number]

Default caption in log files: "Pickup Latitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LAT | UNIT_ANGLE | LOG_VALUE

The actual position (latitude) from which the anchor was recovered. This variable is for anchor handling operations and is designed for use with anchor log files.

A.6.1.23 AHT.Pickup.Pos.Lon

[Real Number]

Default caption in log files: "Pickup Longitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LON | UNIT_ANGLE | LOG_VALUE

The actual position (longitude) from which the anchor was recovered. This variable is for anchor handling operations and is designed for use with anchor log files.

A.6.1.24 AHT.Pos.Lat

[Real Number]

Default caption in log files: "Latitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LAT | UNIT_ANGLE | LOG_VALUE

The last position (latitude) at which an anchor was either dropped or recovered. This variable is for anchor handling operations and is designed for use with anchor log files.

A.6.1.25 AHT.Pos.Lon

[Real Number]

Default caption in log files: "Longitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LON | UNIT_ANGLE | LOG_VALUE

The last position (longitude) at which an anchor was either dropped or recovered. This variable is for anchor handling operations and is designed for use with anchor log files.

A.6.1.26 AHT.Requestor

[String]

Default caption in log files: "Requestor"
Attribute flags: TYPE_STRING

The name of the barge or tug requesting an anchor action. This variable is for anchor handling operations and is designed for use with anchor log files.

A.6.1.27 AHT.Target.Grid.Easting

[Real Number]

Default caption in log files: "Intended Easting"
Default format specifier: "%+.0f"
Attribute flags: TYPE_DOUBLE | TYPE2_EAST | LOG_VALUE | SCAN_NOLOG

The target drop grid position (easting) for the anchor deployment. This variable is for anchor handling operations and is designed for use with anchor log files.

A.6.1.28 AHT.Target.Grid.Northing

[Real Number]

Default caption in log files: "Intended Northing"
Default format specifier: "%+.0f"
Attribute flags: TYPE_DOUBLE | TYPE2_NORTH | LOG_VALUE | SCAN_NOLOG

The target drop grid position (northing) for the anchor deployment. This variable is for anchor handling operations and is designed for use with anchor log files.

A.6.1.29 AHT.Target.Pos.Lat

[Real Number]

Default caption in log files: "Intended Latitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LAT | UNIT_ANGLE | LOG_VALUE

The target drop position (latitude) for the anchor deployment. This variable is for anchor handling operations and is designed for use with anchor log files.

A.6.1.30 AHT.Target.Pos.Lon

[Real Number]

Default caption in log files: "Intended Longitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LON | UNIT_ANGLE | LOG_VALUE

The target drop position (longitude) for the anchor deployment. This variable is for anchor handling operations and is designed for use with anchor log files.

A.6.1.31 AHT.Tug.Name

[String]

Default caption in log files: "Tug Name"
Attribute flags: TYPE_STRING

The name of the tug being referred to in the anchor request. This variable is for anchor handling operations and is designed for use with anchor log files.

A.6.1.32 Alert.Description

[String]

Default caption in log files: "Alert Description"
Attribute flags: TYPE_STRING | SCAN_NOLOG

The last recorded system alert string.

A.6.1.33 AutoPilot.Direction

[String]

Default caption in log files: "Auto Pilot Direction"
Attribute flags: TYPE_STRING | SCAN_SLOWLOG | LOG_VALUE

The direction 'L' or 'R' that an autopilot should steer in order to stay on the active route line.

A.6.1.34 AutoPilot.ReversedDirection

[String]

Default caption in log files: "Reverse Auto Pilot Direction"
Attribute flags: TYPE_STRING | SCAN_SLOWLOG | LOG_VALUE

The direction 'L' or 'R' that an autopilot should steer in order to stay on the active route line if the vessel needs to move in reverse.

A.6.1.35 AutoPilot.SP2.Direction

[String]

Default caption in log files: "SP2 Auto Pilot Direction"
Attribute flags: TYPE_STRING | SCAN_SLOWLOG | LOG_VALUE
Old style name: SP2AutoPilotDirection

The direction 'L' or 'R' that an autopilot (steering SP2) should steer in order to stay on the active route line.

A.6.1.36 AutoPilot.SP2.ReversedDirection

[String]

Default caption in log files: "SP2 Reverse Auto Pilot Direction"
Attribute flags: TYPE_STRING | SCAN_SLOWLOG | LOG_VALUE

The direction 'L' or 'R' that an autopilot (steering SP2) should steer in order to stay on the active route line. if SP2 needs to move in reverse.

A.6.1.37 Beacon.ID

[String]

Default caption in log files: "Beacon ID"
Attribute flags: TYPE_STRING | LOG_VALUE | SCAN_NOLOG

ID of the beacon from which a position was last received. This will be something like "A11" or "B12". RTT inputs are also treated as beacons and if the position is for an RTT channel then then the beacon id will be something like "RTT_01".
Although Beacon.X, Beacon.Y, and Beacon.Z are set to blank after logging the data to the HPR CSV log the Beacon.ID value remains set to the ID of the last beacon received. Beacon.Pos.Lat and Beacon.Pos.Lon remain set as well. Note that none of

the Beacon variables are logged to the SQL database in the varloga or b tables. There is a separate special database table 'usbl_log' for this type of data.

A.6.1.38 Beacon.Pos.Lat

[Real Number]

Default caption in log files: "Beacon Latitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LAT | UNIT_ANGLE | SCAN_NOLOG | LOG_VALUE

The latitude value received from the last HPR or RTT message. This variable is intended to be used in the special HPR CSV log file. See Beacon.ID

A.6.1.39 Beacon.Pos.Lon

[Real Number]

Default caption in log files: "Beacon Longitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LON | UNIT_ANGLE | SCAN_NOLOG | LOG_VALUE

The longitude value received from the last HPR or RTT message. This variable is intended to be used in the special HPR CSV log file. See Beacon.ID

A.6.1.40 Beacon.X

[Real Number]

Default caption in log files: "Beacon X"
Default format specifier: "%+.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_NOLOG

The raw X value received from the last HPR or RTT message. This variable is intended to be used in the special HPR CSV log file. Note: If you attempt to use it in a conventional CSV log you will probably loose data as Normal log files are written at a different rate. When the data is written to a HPR log file it is written to the moment any HPR (or RTT) data arrives. For HPR or RTT data where an easting/northing (e.g UTM) position is received in the raw message the X value will log the actual received easting value and is therefore NOT vessel relative. See Beacon.ID

A.6.1.41 Beacon.Y

[Real Number]

Default caption in log files: "Beacon Y"
Default format specifier: "%+.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_NOLOG

The raw Y value received from the last HPR or RTT message. This variable is intended to be used in the special HPR CSV log file. Note: If you attempt to use it in a conventional CSV log you will probably loose data as Normal log files are written at a different rate. When the data is written to a HPR log file it is written to the moment any HPR (or RTT) data arrives. For HPR or RTT data where an easting/northing position (e.g UTM) is received in the raw message the Y value will log the actual received northing value and is therefore NOT vessel relative.See Beacon.ID See Beacon.ID

A.6.1.42 Beacon.Z

[Real Number]

Default caption in log files: "Beacon dz"
Default format specifier: "%+.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_NOLOG

The raw Z value received from the last HPR or RTT message. This variable is intended to be used in the special HPR CSV log file. Note: If you attempt to use it in a conventional CSV log you will probably loose data as Normal log files are written at a different rate. When the data is written to a HPR log file it is written to the moment any HPR (or RTT) data arrives. See Beacon.ID

A.6.1.43 Cable.AUX1.Length

[Real Number]

Default caption in log files: "AUX 1 Cable Length"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | UNIT2_KM | LOG_VALUE

Cable length in kilometres for the AUX2 cable channel (cable operations).

A.6.1.44 Cable.AUX1.SlackFromSectionStart

[Real Number]

Default caption in log files: "AUX 1 Slack From Section Start"
Default format specifier: "%.1f"
Attribute flags: TYPE_DOUBLE | UNIT_RATIO | UNIT2_PERCENT | LOG_VALUE

Smoothed slack percentage from section start for AUX1 cable channel (cable operations).

A.6.1.45 Cable.AUX1.Smoothed.Slack

[Real Number]

Default caption in log files: "AUX 1 Smoothed Slack"
Default format specifier: "%.1f"
Attribute flags: TYPE_DOUBLE | UNIT_RATIO | UNIT2_PERCENT | LOG_VALUE

For cable operations. Smoothed slack percentage for AUX1 cable channel (cable operations).

A.6.1.46 Cable.AUX1.Smoothed.Speed

[Real Number]

Default caption in log files: "AUX 1 Cable Speed"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | UNIT_SPEED | UNIT2_KM_H | LOG_VALUE

Cable speed in km/h for the AUX1 cable channel (cable operations).

A.6.1.47 Cable.AUX1.Smoothed.Tension

[Real Number]

Default caption in log files: "AUX 1 Cable Tension"
Default format specifier: "%.1f"
Attribute flags: TYPE_DOUBLE | LOG_VALUE

Cable tension in (kN) for the AUX1 cable channel (cable operations).

A.6.1.48 Cable.AUX1.Speed

[Real Number]

Default caption in log files: "Raw Aux1 Cable Speed"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | UNIT_SPEED | UNIT2_KM_H | LOG_VALUE

Raw cable speed in km/h for the aux1 cable channel (cable operations).

A.6.1.49 Cable.AUX1.Tension

[Real Number]

Default caption in log files: "Raw AUX 1 Cable Tension"
Default format specifier: "%.1f"
Attribute flags: TYPE_DOUBLE | LOG_VALUE

Raw cable tension in (kN) for the AUX1 cable channel (cable operations).

A.6.1.50 Cable.AUX2.Length

[Real Number]

Default caption in log files: "AUX 2 Cable Length"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | UNIT2_KM | LOG_VALUE

Cable length in kilometres for the AUX2 cable channel (cable operations).

A.6.1.51 Cable.AUX2.SlackFromSectionStart

[Real Number]

Default caption in log files: "AUX 2 Slack From Section Start"
Default format specifier: "%.1f"
Attribute flags: TYPE_DOUBLE | UNIT_RATIO | UNIT2_PERCENT | LOG_VALUE

Smoothed slack percentage from section start for AUX2 cable channel (cable operations).

A.6.1.52 Cable.AUX2.Smoothed.Slack

[Real Number]

Default caption in log files: "AUX 2 Smoothed Slack"
Default format specifier: "%.1f"
Attribute flags: TYPE_DOUBLE | LOG_VALUE

Smoothed slack percentage for AUX2 cable channel (cable operations).

A.6.1.53 Cable.AUX2.Smoothed.Speed

[Real Number]

Default caption in log files: "AUX 2 Cable Speed"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | UNIT_SPEED | UNIT2_KM_H | LOG_VALUE

Cable speed in km/h for the AUX2 cable channel (cable operations).

A.6.1.54 Cable.AUX2.Smoothed.Tension

[Real Number]

Default caption in log files: "AUX 2 Cable Tension"
Default format specifier: "%.1f"
Attribute flags: TYPE_DOUBLE | LOG_VALUE

Cable tension in (kN) for the AUX2 cable channel (cable operations).

A.6.1.55 Cable.AUX2.Speed

[Real Number]

Default caption in log files: "Raw Aux2 Cable Speed"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | UNIT_SPEED | UNIT2_KM_H | LOG_VALUE

Raw cable speed in km/h for the aux2 cable channel (cable operations).

A.6.1.56 Cable.AUX2.Tension

[Real Number]

Default caption in log files: "Raw AUX 2 Cable Tension"
Default format specifier: "%.1f"
Attribute flags: TYPE_DOUBLE | LOG_VALUE

Raw cable tension in (kN) for the AUX2 cable channel (cable operations).

A.6.1.57 Cable.ControlSpeed

[Real Number]

Default caption in log files: "Target Cable Speed"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_SPEED | SCAN_SLOWLOG | LOG_VALUE

The speed in m/s being automatically demanded on systems with automatic cable control. For cable operations.

A.6.1.58 Cable.DistanceDeviation

[Real Number]

Default caption in log files: "Distance Deviation"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | SCAN_SLOWLOG | LOG_VALUE

The distance deviation is based on slack and cable speed. For cable operations.

A.6.1.59 Cable.Engine1.CableOut

[Real Number]

Default caption in log files: "Cable Engine 1 Length"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | LOG_VALUE | SCAN_SLOWLOG

Cable out length from cable engine #1

A.6.1.60 Cable.Engine1.Tension

[Real Number]

Default caption in log files: "Cable Engine 1 Tension"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | SCAN_SLOWLOG | LOG_VALUE

Cable tension from cable engine #1

A.6.1.61 Cable.Engine2.CableOut

[Real Number]

Default caption in log files: "Cable Engine 2 Length"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | LOG_VALUE | SCAN_SLOWLOG

Cable out length from cable engine #2

A.6.1.62 Cable.Engine2.Tension

[Real Number]

Default caption in log files: "Cable Engine 2 Tension"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | SCAN_SLOWLOG | LOG_VALUE

Cable tension from cable engine #2

A.6.1.63 Cable.Engine3.CableOut

[Real Number]

Default caption in log files: "Cable Engine 3 Length"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | LOG_VALUE | SCAN_SLOWLOG

Cable out length from cable engine #3

A.6.1.64 Cable.Engine3.Tension

[Real Number]

Default caption in log files: "Cable Engine 3 Tension"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | SCAN_SLOWLOG | LOG_VALUE

Cable tension from cable engine #3

A.6.1.65 Cable.Engine4.CableOut

[Real Number]

Default caption in log files: "Cable Engine 4 Length"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | LOG_VALUE | SCAN_SLOWLOG

Cable out length from cable engine #4

A.6.1.66 Cable.Engine4.Tension

[Real Number]

Default caption in log files: "Cable Engine 4 Tension"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | SCAN_SLOWLOG | LOG_VALUE

Cable tension from cable engine #4

A.6.1.67 Cable.Factory.Length

[Real Number]

Default caption in log files: "Factory Cable Length"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

Factory length of the cable (cable operations).

A.6.1.68 Cable.Grid.Easting

[Real Number]

Default caption in log files: "Cable Easting"
Default format specifier: "%+.2f"
Attribute flags: TYPE_DOUBLE | TYPE2_EAST | SCAN_NOLOG | LOG_VALUE

The grid position of (cable detector) (easting). The position given here is according to your selected map projection and datum.

A.6.1.69 Cable.Grid.Northing

[Real Number]

Default caption in log files: "Cable Northing"
Default format specifier: "%+.2f"
Attribute flags: TYPE_DOUBLE | TYPE2_NORTH | SCAN_NOLOG | LOG_VALUE

The grid position of (cable detector) (northing). The position given here is according to your selected map projection and datum.

A.6.1.70 Cable.PLC1.Raw.Count

[Real Number]

Default caption in log files: "PLCc1 Raw Counts"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | LOG_VALUE | SCAN_SLOWLOG

Encoder counter value from cable engine #1 this value will be scaled by the navslack encoder/strain gauges scaling factors.

A.6.1.71 Cable.PLC1.Raw.Tension

[Real Number]

Default caption in log files: "PLCc1 Raw Tension"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | LOG_VALUE | SCAN_SLOWLOG

Raw load cell tension value from cable engine #1. This value will be scaled by the navslack encoder/strain gauges scaling factors.

A.6.1.72 Cable.PLC2.Raw.Count

[Real Number]

Default caption in log files: "PLCc2 Raw Counts"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | LOG_VALUE | SCAN_SLOWLOG

Encoder counter value from cable engine #2 this value will be scaled by the navslack encoder/strain gauges scaling factors.

A.6.1.73 Cable.PLC2.Raw.Tension

[Real Number]

Default caption in log files: "PLCc2 Raw Tension"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | LOG_VALUE | SCAN_SLOWLOG

Raw load cell tension value from cable engine #2. This value will be scaled by the navslack encoder/strain gauges scaling factors.

A.6.1.74 Cable.PLC3.Raw.Count

[Real Number]

Default caption in log files: "PLCc3 Raw Counts"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | LOG_VALUE | SCAN_SLOWLOG

Encoder counter value from cable engine #3 this value will be scaled by the navslack encoder/strain gauges scaling factors

A.6.1.75 Cable.PLC3.Raw.Tension

[Real Number]

Default caption in log files: "PLCc3 Raw Tension"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | LOG_VALUE | SCAN_SLOWLOG

Raw load cell tension value from cable engine #1. This value will be scaled by the navslack encoder/strain gauges scaling factors

A.6.1.76 Cable.Pos.Alt

[Real Number]

Default caption in log files: "Cable Altitude"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The geodetic position of (cable detector) (altitude). The position given here is in your selected working datum which is not necessarily WGS84.

A.6.1.77 Cable.Pos.Lat

[Real Number]

Default caption in log files: "Cable Latitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LAT | UNIT_ANGLE | LOG_VALUE

The geodetic position of (cable detector) (longitude). The position given here is in your selected working datum which is not necessarily WGS84.

A.6.1.78 Cable.Pos.Lon

[Real Number]

Default caption in log files: "Cable Longitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LON | UNIT_ANGLE | LOG_VALUE

The geodetic position of (cable detector) (latitude). The position given here is in your selected working datum which is not necessarily WGS84.

A.6.1.79 Cable.Primary.Length

[Real Number]

Default caption in log files: "Primary Cable Length"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | UNIT2_KM | LOG_VALUE

Cable length in kilometres for the primary cable channel (cable operations).

A.6.1.80 Cable.Primary.SlackFromSectionStart

[Real Number]

Default caption in log files: "Primary Slack From Section Start"
Default format specifier: "%.1f"
Attribute flags: TYPE_DOUBLE | UNIT_RATIO | UNIT2_PERCENT | LOG_VALUE

Smoothed slack percentage from section start for primary cable channel (cable operations).

A.6.1.81 Cable.Primary.Smoothed.Slack

[Real Number]

Default caption in log files: "Primary Smoothed Slack"
Default format specifier: "%.1f"
Attribute flags: TYPE_DOUBLE | UNIT_RATIO | UNIT2_PERCENT | LOG_VALUE

Smoothed slack percentage for primary cable channel (cable operations).

A.6.1.82 Cable.Primary.Smoothed.Speed

[Real Number]

Default caption in log files: "Primary Cable Speed"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | UNIT_SPEED | UNIT2_KM_H | LOG_VALUE

Smoothed cable speed in km/h for the primary cable channel (cable operations).

A.6.1.83 Cable.Primary.Smoothed.Tension

[Real Number]

Default caption in log files: "Primary Cable Tension"
Default format specifier: "%.1f"
Attribute flags: TYPE_DOUBLE | UNIT_FORCE | LOG_VALUE

Smoothed cable tension in (kN) for the primary cable channel (cable operations).

A.6.1.84 Cable.Primary.Speed

[Real Number]

Default caption in log files: "Raw Primary Cable Speed"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | UNIT_SPEED | UNIT2_KM_H | LOG_VALUE

Raw cable speed in km/h for the primary cable channel (cable operations).

A.6.1.85 Cable.Primary.Tension

[Real Number]

Default caption in log files: "Raw Primary Cable Tension"
Default format specifier: "%.1f"
Attribute flags: TYPE_DOUBLE | LOG_VALUE

Raw cable tension in (kN) for the primary cable channel (cable operations).

A.6.1.86 Cable.RouteDistance

[Real Number]

Default caption in log files: "Ship Route Distance"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | UNIT2_KM | LOG_VALUE

Distance along cable route (cable operations).

A.6.1.87 Cable.TargetSlack

[Real Number]

Default caption in log files: "Target Slack"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | UNIT_RATIO | UNIT2_PERCENT | SCAN_SLOWLOG | LOG_VALUE

The target slack percentage (cable operations)

A.6.1.88 Cable.TargetSpeedKmh

[Real Number]

Default caption in log files: "Desired Cable Speed"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_SPEED | SCAN_SLOWLOG | UNIT2_KM_H | LOG_VALUE

The desired cable speed based on required slack (see Cable.TargetSlack). For cable operations.

A.6.1.89 Cable.TargetTension

[Real Number]

Default caption in log files: "Target Tension"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | UNIT_FORCE | UNIT2_KN | SCAN_SLOWLOG | LOG_VALUE

Target cable tension (cable operations)

A.6.1.90 Clara.AutoSolveMode

[Integer]

Default caption in log files: "Clara Auto Solve Mode"
Default format specifier: "%d"
Attribute flags: TYPE_LONG | SCAN_SLOWLOG | LOG_VALUE

The Clara calculation auto solve mode 0 =no autosolve, 1=use inclinometer angle, 2=use top tension measurement.

A.6.1.91 Clara.CableInfo

[String]

Default caption in log files: "Clara Cable Info"
Default format specifier: "%d"
Attribute flags: TYPE_STRING | SCAN_SLOWLOG | LOG_VALUE

Clara calculation cable name details. This variable holds infomation describing the users cable selection.

A.6.1.92 Clara.MBTension

[Real Number]

Default caption in log files: "Clara Manual Bottom Tension"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_FORCE | SCAN_SLOWLOG | LOG_VALUE

Clara calculation manually set bottom tension. This is NOT a calculation result but the last manual value entered by the user if yension was manually adjusted.

A.6.1.93 Clara.MSeabedSlope

[Real Number]

Default caption in log files: "Clara Manual Seabed Slope"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | SCAN_SLOWLOG | LOG_VALUE

Clara calculation manual seabed slope (entered by user). If the user has manually entered the seabed slope then this variable holds that value.

A.6.1.94 Clara.UserAdjust

[String]

Default caption in log files: "Clara User Adjustment Verb"
Default format specifier: "%d"
Attribute flags: TYPE_STRING | SCAN_NOLOG | LOG_VALUE

Clara calculation adjustment verb (for internal use). This is set by the Clara user interface in Blue Spider whenever the ser makes a manual adjustment. It describes the attempted adjustment.

A.6.1.95 Clara.UseRouteDepth

[Integer]

Default caption in log files: "Clara Using Route Depth"
Default format specifier: "%d"
Attribute flags: TYPE_LONG | SCAN_SLOWLOG | LOG_VALUE

If the Clara calculation uses route slope then this has the value 1.

A.6.1.96 Clara.UseRouteSlope

[Integer]

Default caption in log files: "Clara Using Route Slope"
Default format specifier: "%d"
Attribute flags: TYPE_LONG | SCAN_SLOWLOG | LOG_VALUE

If the Clara calculation uses route slope then this has the value 1.

A.6.1.97 GPS1.Altitude

[Real Number]

Default caption in log files: "GPS1 Altitude"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The altitude reported by the GPS receiver #1. BSPEngine expects all GPS receivers to output position in WGS84 and this means the altitude is also with respect to the WGS84 geoid. The altitude here is the altitude of the antenna not the CRP.

A.6.1.98 GPS1.AltitudeWGS84

[Real Number]

Default caption in log files: "GPS1 Altitude (WGS84)"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The altitude reported by the GPS receiver #1.BSPEngine expects all GPS receivers to

output position in WGS84 and this means the altitude is also with respect to the WGS84 geoid.The altitude here is the altitude of the antenna not the CRP.

A.6.1.99 GPS1.CRP.DX

[Real Number]

Default caption in log files: "GPS1 CRP DX"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_NOLOG

Computed DX value of CRP position computed by GPS1.

A.6.1.100 GPS1.CRP.DY

[Real Number]

Default caption in log files: "GPS1 CRP DY"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_NOLOG

Computed DY value of CRP position computed by GPS1.

A.6.1.101 GPS1.CRP.DZ

[Real Number]

Default caption in log files: "GPS1 CRP DZ"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_NOLOG

Computed DZ value of CRP position computed by GPS1.

A.6.1.102 GPS1.CRP.Pos.Alt

[Real Number]

Default caption in log files: "GPS1 CRP Altitude"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_NOLOG

Computed altitude value of CRP position derived from GPS1 in your working datum/vertical reference

A.6.1.103 GPS1.CRP.Pos.Lat

[Real Number]

Default caption in log files: "GPS1 CRP Latitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LAT | UNIT_ANGLE | LOG_VALUE | SCAN_NOLOG

Computed CRP position (latitude) derived from GPS1 in your working datum

A.6.1.104 GPS1.CRP.Pos.Lon

[Real Number]

Default caption in log files: "GPS1 CRP Longitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LAT | UNIT_ANGLE | LOG_VALUE | SCAN_NOLOG

Computed CRP position (longitude) derived from GPS1 in your working datum

A.6.1.105 GPS1.CRP.WGS84.Pos.Alt

[Real Number]

Default caption in log files: "GPS1 CRP Altitude (WGS84)"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_NOLOG

Computed altitude value of CRP derived from GPS1 in WGS84

A.6.1.106 GPS1.CRP.WGS84.Pos.Lat

[Real Number]

Default caption in log files: "GPS1 CRP Latitude (WGS84)"
Attribute flags: TYPE_DOUBLE | TYPE2_LAT | UNIT_ANGLE | LOG_VALUE | SCAN_NOLOG

Computed CRP position (latitude) derived from GPS1 in WGS84

A.6.1.107 GPS1.CRP.WGS84.Pos.Lon

[Real Number]

Default caption in log files: "GPS1 CRP Longitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LAT | UNIT_ANGLE | LOG_VALUE | SCAN_NOLOG

Computed CRP position (longitude) derived from GPS1 in WGS84

A.6.1.108 GPS1.Date

[String]

Default caption in log files: "GPS1 Date"
Attribute flags: TYPE_STRING | TYPE2_DATE | ASSOC_NEXT | SCAN_NOLOG

The time (date) of GPS1 (DD/MM/YYYY) as received from GPS receiver #1.

A.6.1.109 GPS1.DatumShifted.Pos.Alt

[Real Number]

Default caption in log files: "GPS1 Altitude (DS)"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The position of GPS1 (altitude). The position given here is in your selected working datum / vertical reference which is not necessarily WGS84.

A.6.1.110 GPS1.DatumShifted.Pos.Lat

[Real Number]

Default caption in log files: "GPS1 Latitude (DS)"
Attribute flags: TYPE_DOUBLE | TYPE2_LAT | UNIT_ANGLE | SCAN_SLOWLOG | LOG_VALUE

The geodetic position of GPS1 (latitude). The position given here is in your selected working datum which is not necessarily WGS84.

A.6.1.111 GPS1.DatumShifted.Pos.Lon

[Real Number]

Default caption in log files: "GPS1 Longitude (DS)"
Attribute flags: TYPE_DOUBLE | TYPE2_LON | UNIT_ANGLE | SCAN_SLOWLOG | LOG_VALUE

The geodetic position of GPS1 (longitude). The position given here is in your selected working datum which is not necessarily WGS84.

A.6.1.112 GPS1.GeoidalSeparation

[Real Number]

Default caption in log files: "GPS1 GeoidalSeparation"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The geoidal separation reported by the GPS receiver #1.

A.6.1.113 GPS1.GPS2.Heading

[Real Number]

Default caption in log files: "GPS1-2 Derived Heading"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | LOG_VALUE

Computed raw vessel heading derived from GPS1 to GPS2 vector.

A.6.1.114 GPS1.Grid.Easting

[Real Number]

Default caption in log files: "GPS1 Easting"
Default format specifier: "%+.2f"
Attribute flags: TYPE_DOUBLE | TYPE2_EAST | SCAN_NOLOG | LOG_VALUE

The grid position of GPS1 (easting). The position given here is according to your selected map projection and datum.

A.6.1.115 GPS1.Grid.Northing

[Real Number]

Default caption in log files: "GPS1 Northing"
Default format specifier: "%+.2f"
Attribute flags: TYPE_DOUBLE | TYPE2_NORTH | SCAN_NOLOG | LOG_VALUE

The grid position of GPS1 (northing). The position given here is according to your selected map projection and datum.

A.6.1.116 GPS1.HDOP

[Real Number]

Default caption in log files: "GPS1 HDOP"
Default format specifier: "%.1f"
Attribute flags: TYPE_DOUBLE | LOG_VALUE

The HDOP (horizontal dilution of precision) as received from the GPS receiver #1.

A.6.1.117 GPS1.PDOP

[Real Number]

Default caption in log files: "GPS1 PDOP"
Default format specifier: "%.1f"
Attribute flags: TYPE_DOUBLE | LOG_VALUE

The PDOP (dilution of precision) as received from the GPS receiver #1.

A.6.1.118 GPS1.Pos.Lat

[Real Number]

Default caption in log files: "GPS1 Latitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LAT | UNIT_ANGLE | LOG_VALUE

The geodetic position reported by GPS1 (latitude). The position given here is the raw position in WGS84.

A.6.1.119 GPS1.Pos.Lon

[Real Number]

Default caption in log files: "GPS1 Longitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LON | UNIT_ANGLE | LOG_VALUE

The geodetic position reported by GPS1 (longitude). The position given here is the raw position in WGS84.

A.6.1.120 GPS1.Quality

[String]

Default caption in log files: "GPS1 Quality"
Default format specifier: "%.1f"
Attribute flags: TYPE_STRING

The quality indicator reported by GPS receiver #1.

A.6.1.121 GPS1.Sats

[Integer]

Default caption in log files: "GPS1 Sats"
Default format specifier: "%d"
Attribute flags: TYPE_LONG | LOG_VALUE

The number of satellites/ground stations in view to GPS receiver #1

A.6.1.122 GPS1.Time

[String]

Default caption in log files: "GPS1 Time"
Attribute flags: TYPE_STRING | TYPE2_TIME | ASSOC_PREV

The time (time) of GPS1 (HH:MM:SS.SS) as received from GPS receiver #1.

A.6.1.123 GPS1.VDOP

[Real Number]

Default caption in log files: "GPS1 VDOP"
Default format specifier: "%.1f"
Attribute flags: TYPE_DOUBLE | LOG_VALUE

The VDOP (vertical dilution of precision) as received from the GPS receiver #1.

A.6.1.124 GPS2.GPS3.Heading

[Real Number]

Default caption in log files: "GPS2-3 Derived Heading"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | LOG_VALUE

Computed raw vessel heading derived from GPS2 to GPS3 vector.

A.6.1.125 GPS3.GPS1.Heading

[Real Number]

Default caption in log files: "GPS3-1 Derived Heading"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | LOG_VALUE

Computed raw vessel heading derived from GPS3 to GPS1 vector.

A.6.1.126 Gyro1.Corr.Heading

[Real Number]

Default caption in log files: "GYRO1 Heading"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | LOG_VALUE

The adjusted heading in degrees reported by GYRO #1

A.6.1.127 Gyro1.Heading

[Real Number]

Default caption in log files: "GYRO1 Heading"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | LOG_VALUE

The adjusted heading in degrees reported by GYRO #1

A.6.1.128 Gyro1.Message

[String]

Default caption in log files: "HDT1 msg"
Attribute flags: TYPE_STRING | SCAN_NOLOG

Message string from the gyro #1 input.

A.6.1.129 Gyro1.Raw.Heading

[Real Number]

Default caption in log files: "Raw GYRO1 Heading"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | LOG_VALUE

The raw heading in degrees reported by GYRO #1

A.6.1.130 HPR.Ancilliary.Heading

[Real Number]

Default caption in log files: "Heading (from HPR)"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | LOG_VALUE | SCAN_NOLOG

Heading value from HPR message (only a few HPR message types are likely to have this data)

A.6.1.131 HPR.Ancilliary.Heave

[Real Number]

Default caption in log files: "Heave (from HPR)"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_NOLOG

Heave value from HPR message (only a few HPR message types are likely to have this data)

A.6.1.132 HPR.Ancilliary.Pitch

[Real Number]

Default caption in log files: "Pitch (from HPR)"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | LOG_VALUE | SCAN_NOLOG

Pitch value from HPR message (only a few HPR message types are likely to have this data)

A.6.1.133 HPR.Ancilliary.Roll

[Real Number]

Default caption in log files: "Roll (from HPR)"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | LOG_VALUE | SCAN_NOLOG

Roll value from HPR message (only a few HPR message types are likely to have this data)

A.6.1.134 Logging.Backup1.AnticipatedSize

[String]

Default caption in log files: "Backup Log #1 Anticipated File Size"
Attribute flags: TYPE_STRING | SCAN_NOLOG

Anticipated size of the backup log file #1 in bytes

A.6.1.135 Logging.Backup1.FileSize

[String]

Default caption in log files: "Backup Log #1 Filesize"
Attribute flags: TYPE_STRING | SCAN_NOLOG

Size of the backup log file #1 in bytes

A.6.1.136 Logging.Backup1.Unc

[String]

Default caption in log files: "Backup Log #1 UNC"
Attribute flags: TYPE_STRING | SCAN_SLOWLOG

UNC filename of the backup log file #1

A.6.1.137 Logging.Cable.Line.Name

[Unknown]

Default caption in log files: "Cable Line Name"
Attribute flags: TYPE_UNDEFINED | SCAN_NOLOG

The cable line name DEPRECATED (do not use) will only output empty string.

A.6.1.138 Logging.Cable.Line.No

[String]

Default caption in log files: "Cable Line No"
Attribute flags: TYPE_STRING | LOG_VALUE

The cable line number (string). This is only applicable when NavSlack is being used for cable operations.

A.6.1.139 Logging.Cable.Type

[String]

Default caption in log files: "Cable Type"
Attribute flags: TYPE_STRING

The cable type (string). This is only applicable when NavSlack is being used for cable operations.

A.6.1.140 Logging.Comment

[String]

Default caption in log files: "Comment"
Attribute flags: TYPE_STRING | LOG_TEXT

The fix comment string. This is set when a fix is taken. See also Logging.Description For automatic fixes it will have the value 'Time' or 'Distance'

A.6.1.141 Logging.Config1.LogType

[String]

Default caption in log files: "Log File #1 Type"
Attribute flags: TYPE_STRING | SCAN_SLOWLOG

Configured type of the primary log file #1

A.6.1.142 Logging.Config1.Name

[String]

Default caption in log files: "Log File #1 Name"
Attribute flags: TYPE_STRING | SCAN_NOLOG

Caption name of log file #1

A.6.1.143 Logging.Description

[String]

Default caption in log files: "Description"
Attribute flags: TYPE_STRING | LOG_TEXT

The fix description string. This is set when a fix is taken. See also Logging.Comment. This variable indicates the type of fix. For automatic fixes it will have the value 'Automatic'

A.6.1.144 Logging.EventNo

[Integer]

Default caption in log files: "Event No"
Default format specifier: "%03d"
Attribute flags: TYPE_LONG | LOG_VALUE

The last event number used when writing to CSV log files. The value is updated whenever a NavSlack event is taken by the user (regardless of whether CSV logging is enabled). If you are not using NavSlack then this variable is not used.

A.6.1.145 Logging.FixedSP

[String]

Default caption in log files: "FixedSP"
Attribute flags: TYPE_STRING | LOG_VALUE

The last fix SP when writing to CSV log files. The value is updated whenever a manual fix is performed (regardless of whether CSV logging is enabled). It is blank for automatic fixes but for manual fixes will contain the string "SP1", "SP2" or "SP3"

A.6.1.146 Logging.FixNo

[Integer]

Default caption in log files: "Fix No"
Default format specifier: "%03d"
Attribute flags: TYPE_LONG | LOG_VALUE

The last fix number used when writing to CSV log files. The value is updated whenever a manual or automatic fix is performed (regardless of whether CSV logging is enabled). Automatic fixes are usually time or distance based.

A.6.1.147 Logging.Primary1.AnticipatedSize

[String]

Default caption in log files: "Primary Log #1 Anticipated File Size"
Attribute flags: TYPE_STRING | SCAN_NOLOG

Anticipated size of the primary log file #1 in bytes

A.6.1.148 Logging.Primary1.FileSize

[String]

Default caption in log files: "Primary Log #1 Filesize"
Attribute flags: TYPE_STRING | SCAN_NOLOG

Size of the primary log file #1 in bytes

A.6.1.149 Logging.Primary1.Unc

[String]

Default caption in log files: "Primary Log #1 UNC"
Attribute flags: TYPE_STRING | SCAN_SLOWLOG

UNC filename of the primary log file #1

A.6.1.150 MRU1.Heave

[Real Number]

Default caption in log files: "MRU1 Heave"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The heave reported by motion sensor #1 device (in metres).

A.6.1.151 MRU1.Pitch

[Real Number]

Default caption in log files: "MRU1 Pitch"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | LOG_VALUE

The pitch reported by motion sensor #1 device (in degrees).

A.6.1.152 MRU1.Roll

[Real Number]

Default caption in log files: "MRU1 Roll"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | LOG_VALUE

The roll reported by motion sensor #1 device (in degrees).

A.6.1.153 Option.SpeedGaugeKmh.Max

[Real Number]

Default caption in log files: "Option Speed Gauge Kmh Max"
Attribute flags: TYPE_DOUBLE | UNIT_SPEED | UNIT2_KM_H | SCAN_SLOWLOG | LOG_VALUE

Speed gauge range max. This variable gives the minimum range of the speed gauge in NavSlack

A.6.1.154 Option.SpeedGaugeKmh.Min

[Real Number]

Default caption in log files: "Option Speed Gauge Kmh Min"
Attribute flags: TYPE_DOUBLE | UNIT_SPEED | UNIT2_KM_H | SCAN_SLOWLOG | LOG_VALUE

Speed gauge range min This variable gives the minimum range of the speed gauge in Navslack

A.6.1.155 Option.TensionGaugeKN.Max

[Real Number]

Default caption in log files: "Option Tension Gauge kN Max"
Attribute flags: TYPE_DOUBLE | UNIT_FORCE | UNIT2_KN | SCAN_SLOWLOG | LOG_VALUE

Tension gauge range max This variable gives the minimum range of the tension gauge in Navslack

A.6.1.156 Option.TensionGaugeKN.Min

[Real Number]

Default caption in log files: "Option Tension Gauge kN Min"
Attribute flags: TYPE_DOUBLE | UNIT_FORCE | UNIT2_KN | SCAN_SLOWLOG | LOG_VALUE

Tension gauge range min This variable gives the minimum range of the tension gauge in NavSlack

A.6.1.157 PrimaryGPS.Altitude

[Real Number]

Default caption in log files: "Primary GPS Altitude"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | SCAN_ALIAS | LOG_VALUE

The altitude reported by the primary GPS receiver. BSPEngine expects all GPS receivers to output position in WGS84 and this means the altitude is also with respect to the WGS84 geoid. The altitude here is the altitude of the antenna not the CRP. This variable is an alias of Ship.GPS.AltitudeWGS84

A.6.1.158 PrimaryGPS.AltitudeWGS84

[Real Number]

Default caption in log files: "Primary GPS Altitude (WGS84)"

Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | SCAN_ALIAS | LOG_VALUE

The altitude reported by the primary GPS receiver. BSPEngine expects all GPS receivers to output position in WGS84 and this means the altitude is also with respect to the WGS84 geoid. The altitude here is the altitude of the antenna not the CRP. This variable is an alias of Ship.GPS.AltitudeWGS84

A.6.1.159 PrimaryGPS.GeoidalSeparation

[Real Number]

Default caption in log files: "Primary GPS Geoidal Separation"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | SCAN_ALIAS | LOG_VALUE

The altitude reported by the primary GPS receiver. BSPEngine expects all GPS receivers to output position in WGS84 and this means the altitude is also with respect to the WGS84 geoid. The altitude here is the altitude of the antenna not the CRP. This variable is an alias of Ship.GPS.AltitudeWGS84

A.6.1.160 PrimaryGPS.HDOP

[Real Number]

Default caption in log files: "Primary GPS HDOP"
Default format specifier: "%.1f"
Attribute flags: TYPE_DOUBLE | SCAN_ALIAS | LOG_VALUE

The primary GPS HDOP (horizontal dilution of precision) as received from the primary GPS receiver. This variable is an alias of Ship.GPS.HDOP

A.6.1.161 PrimaryGPS.Quality

[String]

Default caption in log files: "Primary GPS Quality"
Default format specifier: "%.1f"
Attribute flags: TYPE_STRING | SCAN_ALIAS

The quality indicator reported by the primary GPS receiver. This variable is an alias of Ship.GPS.Quality

A.6.1.162 PrimaryGPS.Sats

[Integer]

Default caption in log files: "Primary GPS Sats"
Default format specifier: "%d"
Attribute flags: TYPE_LONG | SCAN_ALIAS | LOG_VALUE

The number of satellites/ground stations in view to the GPS receiver. This variable is an alias of Ship.GPS.Sats

A.6.1.163 Remotes.Vessel1.ID

[String]

Default caption in log files: "Remote Vessel 1 ID"
Attribute flags: TYPE_STRING | SCAN_NOLOG

ID of remote vessel #1

A.6.1.164 Remotes.Vessel1.Name

[String]

Default caption in log files: "Remote Vessel 1 Name"
Attribute flags: TYPE_STRING | SCAN_SLOWLOG

Name of remote vessel #1

A.6.1.165 Remotes.Vessel1.Push.Avg.Latency

[Real Number]

Default caption in log files: "Remote Vessel 1 Receiver Avgd. Push Latency"
Default format specifier: "%.6f"
Attribute flags: TYPE_DOUBLE | UNIT_TIME | SCAN_NOLOG

The averaged amount of time in seconds that it is taking BSPEngine to push status information to remote vessel #1

A.6.1.166 Remotes.Vessel1.Push.Latency

[Real Number]

Default caption in log files: "Remote Vessel 1 Push Latency"
Default format specifier: "%.6f"
Attribute flags: TYPE_DOUBLE | UNIT_TIME | SCAN_NOLOG

The amount of time in seconds that it last took BSPEngine to push status information to remote vessel #1

A.6.1.167 Remotes.Vessel1.SP1.Offset.Name

[String]

Default caption in log files: "Remote Vessel 1 SP1 OffsetName"
Attribute flags: TYPE_STRING | SCAN_SLOWLOG

For remote vessel #1, name of the vessel offset which is currently SP1

A.6.1.168 Remotes.Vessel1.SP1.Pos.Alt

[Real Number]

Default caption in log files: "Remote Vessel 1 SP1 Altitude"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

For remote vessel #1, The altitude of the SP1 offset.

A.6.1.169 Remotes.Vessel1.SP1.Pos.Lat

[Real Number]

Default caption in log files: "Remote Vessel 1 SP1 Latitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LON | UNIT_ANGLE | LOG_VALUE

For remote vessel #1, The latitude of the SP1 offset.

A.6.1.170 Remotes.Vessel1.SP1.Pos.Lon

[Real Number]

Default caption in log files: "Remote Vessel 1 SP1 Longitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LON | UNIT_ANGLE | LOG_VALUE

For remote vessel #1, The longitude of the SP1 offset.

A.6.1.171 Remotes.Vessel1.Stats.Avg.TimeDelta

[Real Number]

Default caption in log files: "Remote Vessel 1 Receiver Avgd. Time Delta"
Default format specifier: "%+.6f"
Attribute flags: TYPE_DOUBLE | UNIT_TIME | SCAN_NOLOG

The measured average time difference in seconds between local and remote vessels system clocks.

A.6.1.172 Remotes.Vessel1.System.Timestamp

[String]

Default caption in log files: "Remote Vessel 1 System Time"
Default format specifier: "*dd/mm/yyyy hh:nn:ss.ss"
Attribute flags: TYPE_STRING | TYPE2_TIME | SCAN_NOLOG

System time and date string from remote vessel #1

A.6.1.173 Remotes.Vessel1.Time.Delta

[Real Number]

Default caption in log files: "Remote Vessel 1 Time Delta"
Default format specifier: "%+.6f"
Attribute flags: TYPE_DOUBLE | UNIT_TIME | SCAN_NOLOG

The measured time difference in seconds between local and remote vessels system clocks.

A.6.1.174 Remotes.Vessel1.Time.Estimated.Timestamp

[String]

Default caption in log files: "Remote Vessel 1 Estimated System Timestamp"
Default format specifier: "*dd/mm/yyyy hh:nn:ss.ss"
Attribute flags: TYPE_STRING | TYPE2_TIME | SCAN_NOLOG

The system time and date string from remote vessel #1 taking account of push latency.

A.6.1.175 Route.Direction

[Integer]

Default caption in log files: "Route Direction"
Default format specifier: "%d"
Attribute flags: TYPE_LONG | SCAN_SLOWLOG | LOG_VALUE

The direction of the currently active route. -1 or +1.

A.6.1.176 Route.Name

[String]

Default caption in log files: "Route Name"
Attribute flags: TYPE_STRING

The name of the active route line

A.6.1.177 Route.Target.Name

[String]

Default caption in log files: "Target Name"

Attribute flags: TYPE_STRING | SCAN_ALIAS

Name of the current target.

A.6.1.178 Route.Target1.Pos.Lat

[Real Number]

Default caption in log files: "Target #1 Latitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LAT | UNIT_ANGLE | SCAN_SLOWLOG | LOG_VALUE

The latitude of the current target #1

A.6.1.179 Route.Target1.Pos.Lon

[Real Number]

Default caption in log files: "Target #1 Longitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LON | UNIT_ANGLE | SCAN_SLOWLOG | LOG_VALUE

The latitude of the current target #1

A.6.1.180 Route.Target1.WGS84.Pos.Lat

[Real Number]

Default caption in log files: "Target #1 Latitude (WGS84)"
Attribute flags: TYPE_DOUBLE | TYPE2_LAT | UNIT_ANGLE | SCAN_NOLOG | LOG_VALUE

The latitude of the current target #1

A.6.1.181 Route.Target1.WGS84.Pos.Lon

[Real Number]

Default caption in log files: "Target #1 Longitude (WGS84)"

Attribute flags: TYPE_DOUBLE | TYPE2_LON | UNIT_ANGLE | SCAN_NOLOG |

LOG_VALUE

The latitude of the current target #1

A.6.1.182 RTT_01.Altitude

[Real Number]

Default caption in log files: "RTT_01 Altitude"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | SCAN_SLOWLOG | LOG_VALUE

The altitude reported by the RTT_01 input.

A.6.1.183 RTT_01.Heading

[Real Number]

Default caption in log files: "RTT 01 Heading"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | LOG_VALUE

The raw heading in degrees reported by RTT #1

A.6.1.184 RTT_01.Pos.Lat

[Real Number]

Default caption in log files: "RTT_01 Latitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LAT | UNIT_ANGLE | SCAN_SLOWLOG | LOG_VALUE

The geodetic position of RTT_01 (latitude). The position given here is in your selected working datum which is not necessarily WGS84.

A.6.1.185 RTT_01.Pos.Lon

[Real Number]

Default caption in log files: "RTT_01 Longitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LON | UNIT_ANGLE | SCAN_SLOWLOG |

LOG_VALUE

The geodetic position of RTT_01 (longitude). The position given here is in your selected working datum which is not necessarily WGS84.

A.6.1.186 RTT_01.WaterDepth

[Real Number]

Default caption in log files: "RTT_01 Water Depth"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | SCAN_SLOWLOG | LOG_VALUE

The water depth reported by the RTT_01 input.

A.6.1.187 Ship.AvgWaterLine

[Real Number]

Default caption in log files: "Ship Average WaterLine Altitude"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The averaged altitude of the ship waterline with respect to the working datum. The value is calculated by taking Ship.CRP.Altitude and subtracting the variable Ship.Draft (if it exists) then adding Ship.Heave. Ship.Draft should normally be defined as an input variable in the INI file. Ship.Heave is already defined and is normally configured to be input from a motion sensor. If there is no heave sensor then Ship.Heave is considered to be 0. heave is positive if the CRP is below the waterline and negative if above.

A.6.1.188 Ship.AvgWaterLineWGS84

[Real Number]

Default caption in log files: "Ship Average WaterLine Altitude (WGS84)"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The averaged altitude of the ship waterline with respect to the WGS84 geoid. The value is calculated by taking Ship.CRP.AltitudeWGS84 and subtracting the

variable Ship.Draft (if it exists) then adding Ship.Heave. Ship.Draft should normally be defined as an input variable in the INI file. Ship.Heave is already defined and is normally configured to be input from a motion sensor. If there is no heave sensor then Ship.Heave is considered to be 0. heave is positive if the CRP is below the waterline and negative if above.

A.6.1.189 Ship.CableEngines.PrimaryChannel

[Integer]

Default caption in log files: "Primary Cable Channel"
Default format specifier: "%d"
Attribute flags: TYPE_LONG | LOG_VALUE

Primary cable engine channel number (cable operations)

A.6.1.190 Ship.CRP.AltitudeWGS84

[Real Number]

Default caption in log files: "Primary GPS CRP Altitude (WGS84)"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The altitude reported by the primary GPS receiver but translated to the CRP position. This translation takes account of the height of the antenna as specified in the vessel definition and also any effective shortening of this height due to pitch and roll of the ship. The altitude recorded by this variable is with respect to the WGS84 geoid.

A.6.1.191 Ship.DesiredSpeedKmh

[Real Number]

Default caption in log files: "Desired Ship Speed (km/h)"
Default format specifier: "%.4f"
Attribute flags: TYPE_DOUBLE | UNIT_SPEED | UNIT2_KM_H | LOG_VALUE

The desired ship speed in kilometres per hour. This is applicable for cable operations where the ship speed is based on cable speed and slack requirements.

A.6.1.192 Ship.Draft

[Real Number]

Default caption in log files: "Ship Draft"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_SLOWLOG

Distance from the keel to the waterline.

A.6.1.193 Ship.EchoSounderDepth

[Real Number]

Default caption in log files: "Ship Echo Sounder Depth"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

Raw water depth reading obtained from the primary echo sounder.

A.6.1.194 Ship.EchoSounderDepth1

[Real Number]

Default caption in log files: "Ship Echo Sounder Depth 1"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

Raw water depth reading obtained from echo sounder #1.

A.6.1.195 Ship.GeoidWaterDepth

[Real Number]

Default caption in log files: "Ship Geoid Water Depth"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

This is water depth relative to the current vertical datum (from primary depth sounder)

A.6.1.196 Ship.GeoidWaterDepth1

[Real Number]

Default caption in log files: "Ship Geoid Water Depth 1"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

This is water depth relative to the current vertical datum (from depth sounder #1)

A.6.1.197 Ship.GPS.AltitudeWGS84

[Real Number]

Default caption in log files: "Primary GPS Altitude (WGS84)"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | SCAN_ALIAS | LOG_VALUE

The altitude reported by the primary GPS receiver. BSPEngine expects all GPS receivers to output position in WGS84 and this means the altitude is also with respect to the WGS84 geoid. The altitude here is the altitude of the antenna not the CRP. This is the sum of the altitude and the geoidal separation

A.6.1.198 Ship.GPS.GeoidalSeparation

[Real Number]

Default caption in log files: "Primary GPS Geoidal Separation"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | SCAN_ALIAS | LOG_VALUE

The geoidal separation reported by the primary GPS receiver.

A.6.1.199 Ship.GPS.HDOP

[Real Number]

Default caption in log files: "Primary GPS HDOP"
Default format specifier: "%.1f"
Attribute flags: TYPE_DOUBLE | SCAN_ALIAS | LOG_VALUE

The primary GPS HDOP (horizontal dilution of precision) as received from the primary GPS receiver.

A.6.1.200 Ship.GPS.Pos.Alt

[Real Number]

Default caption in log files: "Primary GPS Altitude"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The altitude reported by the primary GPS receiver. BSPEngine expects all GPS receivers to output position in MSL and this means the altitude is also with respect to the MSL on WGS84 geoid. The altitude here is the altitude of the antenna not the CRP.

A.6.1.201 Ship.GPS.Pos.Lat

[Real Number]

Default caption in log files: "Primary GPS Latitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LON | UNIT_ANGLE | LOG_VALUE

The latitude reported by the primary (or integrated) GPS receiver.

A.6.1.202 Ship.GPS.Pos.Lon

[Real Number]

Default caption in log files: "Primary GPS Longitude"
Default format specifier: NULL
Attribute flags: TYPE_DOUBLE | TYPE2_LON | UNIT_ANGLE | LOG_VALUE

The longitude reported by the primary (or integrated) GPS receiver.

A.6.1.203 Ship.GPS.Quality

[String]

Default caption in log files: "Primary GPS Quality"
Default format specifier: "%.1f"

Attribute flags: TYPE_STRING | SCAN_ALIAS

The quality indicator reported by the primary GPS receiver.

A.6.1.204 Ship.GPS.ReceiverFlags

[Integer]

Default caption in log files: "GPS Receiver Mode"
Default format specifier: "0x%08x"
Attribute flags: TYPE_LONG | LOG_VALUE | SCAN_SLOWLOG

A set of bitflags describing how the primary GPS system is configured. (by default this is logged in hexdecimal) This is essentially the primary GPS receiver index 0, 1, or 2 in the least significant 3 bits combined with some additional bitflags describing the integration mode
Bitmasks for extracting this data are:
GPS_INDEX_MASK 0x07
GPS_AUTO_PRIMARY 0x08
GPS_INTEG_MASK 0xe0
Masking with the index mask gives the primary GPS receiver. If the GPS_AUTO_PRIMARY bit is set then automatic switching of the primary GPS receiver is enabled Bits set in the GPS_INTEG_MASK area (starting at 0x20) indicate the receivers included in integration e.g.
0x20 = GPS1 (integrated), 0x40 = GPS2 (integrated), 0x80 = GPS3 (integrated) If none of the integrate bits are set then integrated mode is not being used

A.6.1.205 Ship.GPS.Sats

[Integer]

Default caption in log files: "Primary GPS Sats"
Default format specifier: "%d"
Attribute flags: TYPE_LONG | SCAN_ALIAS | LOG_VALUE

The number of satellites/ground stations in view to the main GPS receiver.

A.6.1.206 Ship.GPS.VTG.Course

[Real Number]

Default caption in log files: "VTG Ship Course"
Default format specifier: "%.4f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | SCAN_SLOWLOG | LOG_VALUE

The course as reported by the VTG message (if present) from the primary GPS. Depending on ship speed and the make of GPS receivers being used this speed value may be either more reliable or totally unreliable or somewhere in between. at very low speeds the ship course is always rather erratic.

A.6.1.207 Ship.GPS.VTG.Speed

[Real Number]

Default caption in log files: "VTG Ship Speed"
Default format specifier: "%.4f"
Attribute flags: TYPE_DOUBLE | UNIT_SPEED | UNIT2_KM_H | SCAN_SLOWLOG | LOG_VALUE

The ship speed as reported by the GPS VTG message (if present) from the primary GPS. Depending on ship speed and the make of GPS receivers being used this the speed value may be either more reliable or totally unreliable or somewhere in between.

A.6.1.208 Ship.GridHeading

[Real Number]

Default caption in log files: "Ship Grid Heading"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | LOG_VALUE

Grid heading based on delta easting/northing and not a true direction. This is computed using vessel heading from the primary gyro (possibly adjusted with a fixed calibration offset).

A.6.1.209 Ship.Gyro.ReceiverFlags

[Integer]

Default caption in log files: "Gyro Receiver Mode"

Default format specifier: "0x%08x"
Attribute flags: TYPE_LONG | LOG_VALUE | SCAN_SLOWLOG

A set of bitflags describing how the primary Gyro system is configured. The least significant 3 bits indicate the primary gyro index this can be either 0, or 1 If bit 0x8000 is set then the system gyro has been set to manual mode and the primary index is ignored

A.6.1.210 Ship.Heading

[Real Number]

Default caption in log files: "Ship Heading"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | LOG_VALUE

Vessel heading from the primary gyro (possibly adjusted with a fixed calibration offset).

A.6.1.211 Ship.Kalman.CMG

[Real Number]

Default caption in log files: "Ship Kalman CMG"
Default format specifier: "%.4f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | SCAN_SLOWLOG | LOG_VALUE

The smoothed ship speed in kilometres per hour (calculated using Kalman filter).

A.6.1.212 Ship.Kalman.Pos.Lat

[Real Number]

Default caption in log files: "Ship Kalman Latitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LAT | UNIT_ANGLE | LOG_VALUE

The smoothed postion of the ship as calculated by the Kalman filter.

A.6.1.213 Ship.Kalman.Pos.Lon

[Real Number]

Default caption in log files: "Ship Kalman Longitude"

Attribute flags: TYPE_DOUBLE | TYPE2_LON | UNIT_ANGLE | LOG_VALUE

The smoothed postion of the ship as calculated by the Kalman filter.

A.6.1.214 Ship.Kalman.Speed

[Real Number]

Default caption in log files: "Ship Kalman Speed"
Default format specifier: "%.4f"
Attribute flags: TYPE_DOUBLE | UNIT_SPEED | UNIT2_KM_H | SCAN_SLOWLOG | LOG_VALUE

The smoothed ship speed in kilometres per hour (calculated using Kalman filter).

A.6.1.215 Ship.KeelHeight

[Real Number]

Default caption in log files: "Ship Keel Height"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | SCAN_NOLOG

The distance from the keel of the ship from the CRP.

A.6.1.216 Ship.LaybackPoint

[String]

Default caption in log files: "Ship Layback Offset Name"
Attribute flags: TYPE_STRING

The name of the vessel offset which is currently the default layback offset for all mobiles

A.6.1.217 Ship.Motion.Heading

[Real Number]

Default caption in log files: "Ship Heading (from MRU)"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | LOG_VALUE

The heading of the vessel (in degrees) as obtained from a motion sensor. (NOT THE GYRO)

A.6.1.218 Ship.Motion.Heave

[Real Number]

Default caption in log files: "Ship Heave"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The heave of the vessel (in metres) as obtained from a motion sensor.

A.6.1.219 Ship.Motion.Pitch

[Real Number]

Default caption in log files: "Ship Pitch"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | LOG_VALUE

The pitch of the vessel (in degrees) as obtained from a motion sensor.

A.6.1.220 Ship.Motion.Roll

[Real Number]

Default caption in log files: "Ship Roll"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | LOG_VALUE

The roll of the vessel (in degrees) as obtained from a motion sensor.

A.6.1.221 Ship.MRU.ReceiverFlags

[Integer]

Default caption in log files: "MRU Receiver Mode"
Default format specifier: "0x%08x"
Attribute flags: TYPE_LONG | LOG_VALUE | SCAN_SLOWLOG

A set of bitflags describing how the primary MRU system is configured. The least significant 3 bits indicate the primary MRU index this can be either 0, 1 or 2 All other bits are reserved for possible future extensions

A.6.1.222 Ship.Offsets.Grid1.Easting

[Real Number]

Default caption in log files: "Offset 1 Easting"
Default format specifier: "%+.3f"
Attribute flags: TYPE_DOUBLE | TYPE2_EAST | SCAN_NOLOG | LOG_VALUE

Position (easting) of vessel offset #1. Offset #1 is the CRP.

A.6.1.223 Ship.Offsets.Grid1.Northing

[Real Number]

Default caption in log files: "Offset 1 Northing"
Default format specifier: "%+.3f"
Attribute flags: TYPE_DOUBLE | TYPE2_NORTH | SCAN_NOLOG | LOG_VALUE

Position (northing) of vessel offset #1. Offset #1 is the CRP.

A.6.1.224 Ship.Offsets.Pos1.Alt

[Real Number]

Default caption in log files: "Offset 1 Altitude"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | SCAN_SLOWLOG | LOG_VALUE

Position (altitude) of vessel offset #1. Offset #1 is the CRP. The altitude given here is in your selected working datum (vertical reference) which is not necessarily WGS84.

A.6.1.225 Ship.Offsets.Pos1.Elev

[Real Number]

Default caption in log files: "Offset 1 Elevation"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | SCAN_SLOWLOG | LOG_VALUE

Elevation from seabed to vessel offset. Offset #1 is the CRP. The elevation given here is the water depth in metres from the seabed to the given vessel offset. An adjustment is made from the position of the echo sounder to the vessel offset by taking account of pitch and roll of the ship. The seabed is assumed to be flat as the SP offset could be some distance away from the echo sounder which will mean that the depth given here is not truly accurate.

A.6.1.226 Ship.Offsets.Pos1.Lat

[Real Number]

Default caption in log files: "Offset 1 Latitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LAT | UNIT_ANGLE | SCAN_SLOWLOG | LOG_VALUE

Position (latitude) of vessel offset #1. Offset #1 is the CRP.

A.6.1.227 Ship.Offsets.Pos1.Lon

[Real Number]

Default caption in log files: "Offset 1 Longitude"

Attribute flags: TYPE_DOUBLE | TYPE2_LON | UNIT_ANGLE | SCAN_SLOWLOG | LOG_VALUE

Position (longitude) of vessel offset #1. Offset #1 is the CRP.

A.6.1.228 Ship.Offsets.WGS84.Pos1.Alt

[Real Number]

Default caption in log files: "Offset 1 Altitude (WGS84)"

Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | SCAN_SLOWLOG | LOG_VALUE

Position (altitude) of vessel offset #1 converted to WGS84. Offset #1 is the CRP.

A.6.1.229 Ship.Offsets.WGS84.Pos1.Lat

[Real Number]

Default caption in log files: "Offset 1 Latitude (WGS84)"
Attribute flags: TYPE_DOUBLE | TYPE2_LAT | UNIT_ANGLE | SCAN_NOLOG | LOG_VALUE

Position (latitude) of vessel offset #1 converted to WGS84. Offset #1 is the CRP.

A.6.1.230 Ship.Offsets.WGS84.Pos1.Lon

[Real Number]

Default caption in log files: "Offset 1 Longitude (WGS84)"
Attribute flags: TYPE_DOUBLE | TYPE2_LON | UNIT_ANGLE | SCAN_NOLOG | LOG_VALUE

Position (longitude) of vessel offset #1 converted to WGS84. Offset #1 is the CRP.

A.6.1.231 Ship.PrimaryGyro.Message

[String]

Default caption in log files: "HDTPrimary msg"
Attribute flags: TYPE_STRING | SCAN_NOLOG

Message string from the primary gyro input.

A.6.1.232 Ship.RawSpeedKmh

[Real Number]

Default caption in log files: "Raw Ship Speed (km/h)"
Default format specifier: "%.4f"
Attribute flags: TYPE_DOUBLE | UNIT_SPEED | UNIT2_KM_H | LOG_VALUE

The raw ship speed in kilometres per hour.

A.6.1.233 Ship.SP1.Grid.Easting

[Real Number]

Default caption in log files: "SP1 Easting"
Default format specifier: "%+.2f"
Attribute flags: TYPE_DOUBLE | TYPE2_EAST | SCAN_NOLOG | LOG_VALUE

The grid position of SP1 (easting). The position given here is according to your selected map projection and datum.

A.6.1.234 Ship.SP1.Grid.Northing

[Real Number]

Default caption in log files: "SP1 Northing"
Default format specifier: "%+.2f"
Attribute flags: TYPE_DOUBLE | TYPE2_NORTH | SCAN_NOLOG | LOG_VALUE

The grid position of SP1 (northing). The position given here is according to your selected map projection and datum.

A.6.1.235 Ship.SP1.Pos.Alt

[Real Number]

Default caption in log files: "SP1 Altitude"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The altitude position of SP1. The altitude given here is in your selected working datum (vertical reference) which is not necessarily WGS84.

A.6.1.236 Ship.SP1.Pos.Elev

[Real Number]

Default caption in log files: "SP1 Elevation"

Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The elevation position of SP1 from the seabed. The elevation given here is the water depth in metres from the seabed to the SP1 vessel offset. An adjustment is made from the position of the echo sounder to the vessel offset of the current SP by taking account of pitch and roll of the ship. The seabed is assumed to be flat as the SP offset could be some distance away from the echo sounder which will mean that the depth given here is not truly accurate.

A.6.1.237 Ship.SP1.Pos.Lat

[Real Number]

Default caption in log files: "SP1 Latitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LAT | UNIT_ANGLE | LOG_VALUE

The geodetic position of SP1 (latitude). The position given here is in your selected working datum which is not necessarily WGS84.

A.6.1.238 Ship.SP1.Pos.Lon

[Real Number]

Default caption in log files: "SP1 Longitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LON | UNIT_ANGLE | LOG_VALUE

The geodetic position of SP1 (longitude).The position given here is in your selected working datum which is not necessarily WGS84.

A.6.1.239 Ship.SP1.Route.Arc.DOL

[Real Number]

Default caption in log files: "SP1 Route Arc DOL"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_NOLOG

The route arc DOL value of SP1. This is the perpendicular distance from route line. On the outer corner of an alter course it is the distance from the corner/arc Alter course radii (if present) are used in this calculation.

A.6.1.240 Ship.SP1.Route.Arc.KP

[Real Number]

Default caption in log files: "SP1 Route Arc Kp"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | UNIT2_KM | LOG_VALUE | SCAN_NOLOG

The route arc KP value of SP1. This is the straight line distance along the route. If no route is active this variable value is undefined (NAN/blank). Alter course radii (if present) are used in this calculation.

A.6.1.241 Ship.SP1.Route.DOL

[Real Number]

Default caption in log files: "SP1 Route DCC"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_NOLOG

The route DOL value of SP1. This is the perpendicular distance from route line. On the outer corner of an alter course it is the distance from the corner (in which case the KP value will be the KP of the corner). Alter course radii are ignored by this calculation.

A.6.1.242 Ship.SP1.Route.Grid.DOL

[Real Number]

Default caption in log files: "SP1 Route GRID DOL"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_NOLOG

The route GRID DOL value of SP1. This is the perpendicular distance from route line. On the outer corner of an alter course it is the distance from the corner (in which case the KP value will be the KP of the corner). Alter course radii are ignored by this calculation and this is a grid calculation not a true KP.

A.6.1.243 Ship.SP1.Route.Grid.KP

[Real Number]

Default caption in log files: "SP1 Route GRID Kp"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | UNIT2_KM | LOG_VALUE | SCAN_NOLOG

The route GRID KP value of SP1. This is the straight line distance along the route. If no route is active this variable value is undefined (NAN/blank). Alter course radii are ignored by this calculation and this is a grid calculation not a true KP.

A.6.1.244 Ship.SP1.Route.KP

[Real Number]

Default caption in log files: "SP1 Route Kp"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | UNIT2_KM | LOG_VALUE | SCAN_NOLOG

The route KP value of SP1. This is the straight line distance along the route. If no route is active this variable value is undefined (NAN/blank). Alter course radii are ignored by this calculation.

A.6.1.245 Ship.Speed

[Real Number]

Default caption in log files: "Ship Speed"
Default format specifier: "%.4f"
Attribute flags: TYPE_DOUBLE | UNIT_SPEED | UNIT2_KM_H | SCAN_SLOWLOG | LOG_VALUE

The smoothed ship speed in kilometres per hour.

A.6.1.246 Ship.SpeedKmh

[Real Number]

Default caption in log files: "Ship Speed (km/h)"
Default format specifier: "%.4f"

Attribute flags: TYPE_DOUBLE | UNIT_SPEED | UNIT2_KM_H | SCAN_ALIAS | LOG_VALUE

The ship speed in kilometres per hour. Alias for Ship.Speed

A.6.1.247 Ship.SpeedMS

[Real Number]

Default caption in log files: "Ship Speed (m/s)"
Default format specifier: "%.4f"
Attribute flags: TYPE_DOUBLE | UNIT_SPEED | SCAN_UNIT_ALIAS | LOG_VALUE
Old style name: ShipSpeedMS

The smoothed ship speed in metres per second.

A.6.1.248 Ship.VDatumShift

[Real Number]

Default caption in log files: "Ship Vert. Datum Shift"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | SCAN_SLOWLOG

The vertical datum shift to convert from the WGS84 geoid altitude to your selected vertical reference. The value given by this variable is valid only for the positions very close to the ship (it is computed for the CRP). The vertical shift required will typically depend on position. If you need to add an additional fixed amount to this then **DO NOT** ~~use System.VDatumShift (typically defined as an input variable) and add its value as well. See also System.VDatumShift, SP1.VDatumShift, SP2.VDatumShift~~
IMPORTANT NOTE
System.VDatumShift has now been DEPRECATED! Although still supported it will cause an alarm to be raised if set to a non-zero value. Instead you should edit geocalc.dat and add the appropriate tags to a copy of a suitable horizontal datum definition. See Ticket #47

A.6.1.249 Ship.WaterDepth

[Real Number]

Default caption in log files: "Ship Water Depth"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

True water depth from the waterline. (from primary depth sounder) This is water depth from the echo sounder adjusted to CRP level then compenstated for draft and heave

A.6.1.250 Ship.WaterDepth1

[Real Number]

Default caption in log files: "Ship Water Depth 1"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

True water depth from the waterline (from depth sounder #1) This is water depth from the echo sounder adjusted to CRP level then compenstated for draft and heave

A.6.1.251 Ship.WaterLine

[Real Number]

Default caption in log files: "Ship WaterLine Altitude"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The altitude of the ship waterline with respect to the working datum / vertical reference geoid. The value is calculated by taking Ship.WaterLineWGS84 and datum shifting to the working datum.

A.6.1.252 Ship.WaterLineWGS84

[Real Number]

Default caption in log files: "Ship WaterLine Altitude (WGS84)"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The altitude of the ship waterline with respect to the WGS84 geoid. The value is

calculated by taking Ship.CRP.AltitudeWGS84 and subtracting the variable Ship.Draft (if it exists) then adding Ship.Heave. Ship.Draft should normally be defined as an input variable in the INI file. Ship.Heave is already defined and is normally configured to be input from a motion sensor. If there is no heave sensor then Ship.Heave is considered to be 0. heave is positive if the CRP is below the waterline and negative if above.

A.6.1.253 SP1.Averaged.CMG

[Real Number]

Default caption in log files: "SP1 CMG"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | SCAN_SLOWLOG | LOG_VALUE

The time averaged SP1 course made good in degrees.

A.6.1.254 SP1.Averaged.Speed

[Real Number]

Default caption in log files: "SP1 SpeedMS"
Default format specifier: "%.4f"
Attribute flags: TYPE_DOUBLE | UNIT_SPEED | SCAN_UNIT_ALIAS | SCAN_SLOWLOG | LOG_VALUE

The time averaged SP1 speed in metres per second.

A.6.1.255 SP1.Averaged.SpeedKmh

[Real Number]

Default caption in log files: "SP1 Speed"
Default format specifier: "%.4f"
Attribute flags: TYPE_DOUBLE | UNIT_SPEED | UNIT2_KM_H | SCAN_SLOWLOG | LOG_VALUE

The time averaged SP1 speed in kilometres per hour.

A.6.1.256 SP1.Date

[String]

Default caption in log files: "Overboard Point Date"
Attribute flags: TYPE_STRING | TYPE2_DATE | ASSOC_PREV | SCAN_NOLOG

The time (date) of SP1/GPS (DD/MM/YYYY) as received from the primary GPS receiver.

A.6.1.257 SP1.GPS.AltitudeWGS84

[Real Number]

Default caption in log files: "Primary GPS Altitude (WGS84)"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The altitude reported by the primary GPS receiver. !BSPEngine expects all GPS receivers to output position in WGS84 and this means the altitude is also with respect to the WGS84 geoid. The altitude here is the altitude of the antenna not the CRP. This variable is an alias of Ship.GPS.AltitudeWGS84

A.6.1.258 SP1.GPS.GeoidalSeparation

[Real Number]

Default caption in log files: "Primary GPS Geoidal Separation"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The altitude reported by the primary GPS receiver. !BSPEngine expects all GPS receivers to output position in WGS84 and this means the altitude is also with respect to the WGS84 geoid. The altitude here is the altitude of the antenna not the CRP. This variable is an alias of Ship.GPS.AltitudeWGS84

A.6.1.259 SP1.GPS.HDOP

[Real Number]

Default caption in log files: "Primary GPS HDOP"
Default format specifier: "%.1f"
Attribute flags: TYPE_DOUBLE | LOG_VALUE

The primary GPS HDOP (horizontal dilution of precision) as received from the primary GPS receiver. This variable is an alias of Ship.GPS.HDOP

A.6.1.260 SP1.GPS.PDOP

[Real Number]

Default caption in log files: "Primary GPS PDOP"
Default format specifier: "%.1f"
Attribute flags: TYPE_DOUBLE | LOG_VALUE

The primary GPS PDOP (primary dilution of precision) as received from the primary GPS receiver.

A.6.1.261 SP1.GPS.Quality

[String]

Default caption in log files: "Primary GPS Quality"
Default format specifier: "%.1f"
Attribute flags: TYPE_STRING | LOG_VALUE

The quality indicator reported by the primary GPS receiver. This variable is an alias of Ship.GPS.Quality

A.6.1.262 SP1.GPS.Sats

[Integer]

Default caption in log files: "Primary GPS Sats"
Default format specifier: "%d"
Attribute flags: TYPE_LONG | LOG_VALUE

The number of satellites/ground stations in view to the GPS receiver. This variable is an alias of Ship.GPS.Sats

A.6.1.263 SP1.GPS.VDOP

[Real Number]

Default caption in log files: "Primary GPS VDOP"
Default format specifier: "%.1f"
Attribute flags: TYPE_DOUBLE | LOG_VALUE

The primary GPS PDOP (primary dilution of precision) as received from the primary GPS receiver.

A.6.1.264 SP1.Grid.Easting

[Real Number]

Default caption in log files: "Overboard Point Easting"
Default format specifier: "%+.2f"
Attribute flags: TYPE_DOUBLE | TYPE2_EAST | SCAN_ALIAS | LOG_VALUE

The grid position of SP1 (easting). The position given here is according to your selected map projection and datum.

A.6.1.265 SP1.Grid.Northing

[Real Number]

Default caption in log files: "Overboard Point Northing"
Default format specifier: "%+.2f"
Attribute flags: TYPE_DOUBLE | TYPE2_NORTH | SCAN_ALIAS | LOG_VALUE

The grid position of SP1 (northing). The position given here is according to your selected map projection and datum.

A.6.1.266 SP1.KP

[Real Number]

Default caption in log files: "Overboard Point kp"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | TYPE2_KP | UNIT_DISTANCE | UNIT2_KM | SCAN_ALIAS | LOG_VALUE

The route KP value of SP1. This is the straight line distance along the route. If no route is active this variable value is undefined (NAN/blank). Note: Route KP calculation uses Rhumb line distances

A.6.1.267 SP1.Offset.Name

[String]

Default caption in log files: "SP1 Offset Name"
Attribute flags: TYPE_STRING

The name of the vessel offset which is currently SP1

A.6.1.268 SP1.Pos.Alt

[Real Number]

Default caption in log files: "SP1 Altitude"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_ALIAS

The altitude position of SP1 (Alias for Ship.SP1.Pos.Alt). The altitude given here is in your selected working datum (vertical reference) which is not necessarily WGS84.

A.6.1.269 SP1.Pos.Lat

[Real Number]

Default caption in log files: "Overboard Point Latitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LAT | UNIT_ANGLE | SCAN_ALIAS | LOG_VALUE

The geodetic position of SP1 (latitude). The position given here is in your selected working datum which is not necessarily WGS84.

A.6.1.270 SP1.Pos.Lon

[Real Number]

Default caption in log files: "Overboard Point Longitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LON | UNIT_ANGLE | SCAN_ALIAS | LOG_VALUE

The geodetic position of SP1 (longitude). The position given here is in your selected working datum which is not necessarily WGS84.

A.6.1.271 SP1.Route.DOL

[Real Number]

Default caption in log files: "Overboard Point DCC"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | SCAN_ALIAS | LOG_VALUE

The route DOL value of SP1. This is the perpendicular distance from route line. On the outer corner of an alter course it is the distance from the corner (in which case the KP value will be the KP of the corner). Alter course radii are ignored by this calculation

A.6.1.272 SP1.Route.Grid.DOL

[Real Number]

Default caption in log files: "Overboard Point Grid DOL"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | SCAN_ALIAS | LOG_VALUE

The route GRID DOL value of SP1.This is the perpendicular distance from route line.On the outer corner of an alter course it is the distance from the corner(in which case the KP value will be the KP of the corner).Alter course radii are ignored by this calculation

A.6.1.273 SP1.Route.Grid.KP

[Real Number]

Default caption in log files: "Nav GRID kp"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | TYPE2_KP | UNIT_DISTANCE | UNIT2_KM | SCAN_ALIAS | LOG_VALUE

The route GRID KP value of SP1. This is the straight line distance along the route. If no route is active this variable value is undefined (NAN/blank). This variable is an alias of Ship.SP1.Grid.KP Alter course radii are ignored by this calculation Note: The calculation here uses cartesian distances between easting and northing coordinates

so depending on the distortion of the map projection the distances may be potentially quite different from true distances on the ground

A.6.1.274 SP1.Route.KP

[Real Number]

Default caption in log files: "Nav kp"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | TYPE2_KP | UNIT_DISTANCE | UNIT2_KM | SCAN_ALIAS | LOG_VALUE

The route KP value of SP1. This is the straight line distance along the route. If no route is active this variable value is undefined (NAN/blank). This variable is an alias of SP1.KP Alter course radii are ignored by this calculation

A.6.1.275 SP1.Route.SeabedSlope

[Real Number]

Default caption in log files: "SP1 Route (survey) Seabed Slope"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | SCAN_NOLOG | LOG_VALUE

The route KP (as surveyed) seabed slope value under SP1. The data here comes from the water depth information in the active route (if present). The value is otherwise undefined

A.6.1.276 SP1.Route.Section.Bearing

[Real Number]

Default caption in log files: "SP1 Route Section Bearing"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | SCAN_SLOWLOG | UNIT2_DEGREES | LOG_VALUE

The true bearing of the current route section.

A.6.1.277 SP1.Route.Target.Bearing

[Real Number]

Default caption in log files: "Target Bearing"
Default format specifier: "%.1f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | SCAN_ALIAS | LOG_VALUE

The true bearing in degrees from SP1 to the current target.

A.6.1.278 SP1.Route.Target.Range

[Real Number]

Default caption in log files: "Target Range"
Default format specifier: "%.1f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | SCAN_ALIAS | LOG_VALUE

The range in metres from SP1 to the current target.

A.6.1.279 SP1.Route.TerrainDist

[Real Number]

Default caption in log files: "SP1 Terrain Dist (km)"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | TYPE2_KP | UNIT_DISTANCE | UNIT2_KM | LOG_VALUE

The route terrain distance value of SP1. This is similar to the KP value except the undulations in the route are taken into account. The route terrain distances are calculated by taking the XYZ distances between each point (the route should contain water depth information for this to be effective). Each route point is converted from lat, long, altitude in order to calculate the distance along each route section. The terrain distance for SP1 is therefore effectively the distance of SP1 along this undulating route. If no route is active this variable value is undefined (NAN/blank). Alter course radii are ignored by this calculation NOTE that the active route line should not only contain water depth information (which is used to give an altitude value for each point) but the waypoints should ideally be spaced close enough together such that the curvature of the earth is not of any great significance. This is because the cartesian distance between each waypoint is used to compute the distance. If waypoints are too far apart then the straight line between each may be

sufficiently long to dip well below the seabed. This will mean that the terrain distances will be shorter that may have been expected.

A.6.1.280 SP1.Route.WaterDepth

[Real Number]

Default caption in log files: "SP1 Route (survey) Water Depth"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | SCAN_NOLOG | LOG_VALUE

The route KP (as surveyed) water depth value SP1. The data here comes from the water depth information in the active route (if present). The value is otherwise undefined

A.6.1.281 SP1.Smoothed.CMG

[Real Number]

Default caption in log files: "SP1 CMG"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | SCAN_SLOWLOG | LOG_VALUE
Old style name: SP1SmoothedCMG

The smoothed SP1 course made good in degrees.

A.6.1.282 SP1.Speed

[Real Number]

Default caption in log files: "SP1 SpeedMS"
Default format specifier: "%.4f"
Attribute flags: TYPE_DOUBLE | UNIT_SPEED | SCAN_UNIT_ALIAS | SCAN_SLOWLOG | LOG_VALUE

The smoothed SP1 speed in metres per second.

A.6.1.283 SP1.SpeedKmh

[Real Number]

Default caption in log files: "SP1 Speed"
Default format specifier: "%.4f"
Attribute flags: TYPE_DOUBLE | UNIT_SPEED | UNIT2_KM_H | SCAN_SLOWLOG | LOG_VALUE

The smoothed SP1 speed in kilometres per hour.

A.6.1.284 SP1.Target1.Bearing

[Real Number]

Default caption in log files: "SP1 Target1 Bearing"
Default format specifier: "%.1f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | LOG_VALUE

The true bearing in degrees from SP1 to the auxiliary target #1

A.6.1.285 SP1.Target1.Range

[Real Number]

Default caption in log files: "SP1 Target1 Range"
Default format specifier: "%.1f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The range in metres from SP1 to the auxilliary target #1

A.6.1.286 SP1.Time

[String]

Default caption in log files: "Overboard Point Time"
Attribute flags: TYPE_STRING | TYPE2_TIME | ASSOC_NEXT | SCAN_NOLOG

The time of SP1/GPS (HH:MM:SS.SS) as received from the primary GPS receiver.

A.6.1.287 SP1.WGS84.Pos.Alt

[Real Number]

Default caption in log files: "SP1 Altitude (WGS84)"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The altitude position of SP1 in WGS84.

A.6.1.288 SP1.WGS84.Pos.Lat

[Real Number]

Default caption in log files: "SP1 Latitude (WGS84)"
Attribute flags: TYPE_DOUBLE | TYPE2_LAT | UNIT_ANGLE | LOG_VALUE

The geodetic position of SP1 (latitude). The position given here is in WGS84.

A.6.1.289 SP1.WGS84.Pos.Lon

[Real Number]

Default caption in log files: "SP1 Latitude WGS84"
Attribute flags: TYPE_DOUBLE | TYPE2_LON | UNIT_ANGLE | LOG_VALUE

The geodetic position of SP1 (longitude). The position given here is in WGS84.

A.6.1.290 SP2.Grid.Easting

[Real Number]

Default caption in log files: "SP2 Easting"
Default format specifier: "%+.2f"
Attribute flags: TYPE_DOUBLE | TYPE2_EAST | SCAN_NOLOG | LOG_VALUE

The grid position of SP2 (easting). The position given here is according to your selected map projection and datum.

A.6.1.291 SP2.Grid.Northing

[Real Number]

Default caption in log files: "SP2 Northing"
Default format specifier: "%+.2f"
Attribute flags: TYPE_DOUBLE | TYPE2_NORTH | SCAN_NOLOG | LOG_VALUE

The grid position of SP2 (northing). The position given here is according to your selected map projection and datum.

A.6.1.292 SP2.GridHeading

[Real Number]

Default caption in log files: "SP2 Grid Heading"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | LOG_VALUE

Grid heading based on delta easting/northing and not a true direction. This is computed using mobile heading obtained the vehicle gyro or other source unless SP2 is an offset on the ship in which case the ship heading is used.

A.6.1.293 SP2.Heading

[Real Number]

Default caption in log files: "SP2 Heading"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | LOG_VALUE

The true heading for SP2.

A.6.1.294 SP2.LaybackBearing

[Real Number]

Default caption in log files: "SP2 Layback Bearing"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | LOG_VALUE | SCAN_NOLOG

The layback bearing of SP2 from the Ship.LaybackPoint

A.6.1.295 SP2.LaybackDistance

[Real Number]

Default caption in log files: "SP2 Layback Distance"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_NOLOG

The layback distance of SP2 from the Ship.LaybackPoint

A.6.1.296 SP2.LaybackMode

[String]

Default caption in log files: "SP2 Layback Mode"
Attribute flags: TYPE_STRING

The layback mode of the vehicle which is currently SP2

A.6.1.297 SP2.Motion.Pitch

[Real Number]

Default caption in log files: "SP2 Pitch"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | LOG_VALUE

The pitch of SP2 (in degrees) as obtained from a motion sensor. If SP2 is a vessel offset then this variable has the same value as Ship.Motion.Pitch otherwise if SP2 is a mobile the variable has the value of the mobile pitch or zero if this is not known.

A.6.1.298 SP2.Motion.Roll

[Real Number]

Default caption in log files: "SP2 Roll"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | LOG_VALUE

The roll of SP2 (in degrees) as obtained from a motion sensor. If SP2 is a vessel offset then this variable has the same value as Ship.Motion.Roll otherwise if SP2 is a mobile the variable has the value of the mobile roll or zero if this is not known.

A.6.1.299 SP2.Name

[String]

Default caption in log files: "SP2 Name"
Attribute flags: TYPE_STRING

The name of the vehicle (or vessel offset) which is currently SP2

A.6.1.300 SP2.Offset.Pos.Name

[String]

Default caption in log files: "SP2 Pos Offset Name"
Attribute flags: TYPE_STRING

The name of the vehicle offset which is the positioning offset for SP2 (blank if a vessel offset is used)

A.6.1.301 SP2.Offset.SP.Name

[String]

Default caption in log files: "SP2 SP Offset Name"
Attribute flags: TYPE_STRING

The name of the vehicle offset which is the steer point offset for SP3 (blank if a vessel offset is used)

A.6.1.302 SP2.Offsets.Grid1.Easting

[Real Number]

Default caption in log files: "SP2 Offset 1 Easting"
Default format specifier: "%+.3f"
Attribute flags: TYPE_DOUBLE | TYPE2_EAST | SCAN_NOLOG | LOG_VALUE

Position (easting) of SP2 offset #1. Offset #1 is the CRP.

A.6.1.303 SP2.Offsets.Grid1.Northing

[Real Number]

Default caption in log files: "SP2 Offset 1 Northing"
Default format specifier: "%+.3f"
Attribute flags: TYPE_DOUBLE | TYPE2_NORTH | SCAN_NOLOG | LOG_VALUE

Position (northing) of SP2 offset #1. Offset #1 is the CRP.

A.6.1.304 SP2.Offsets.Pos1.Alt

[Real Number]

Default caption in log files: "SP2 Offset 1 Altitude"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | SCAN_SLOWLOG | LOG_VALUE

Position (altitude) of SP2 (mobile or ship) offset #1. Offset #1 is the CRP. The altitude given here is in your selected working datum (vertical reference) which is not necessarily WGS84.

A.6.1.305 SP2.Offsets.Pos1.Elev

[Real Number]

Default caption in log files: "SP2 Offset 1 Elevation"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | SCAN_SLOWLOG | LOG_VALUE

Elevation from seabed to mobile/vessel offset. Offset #1 is the CRP. The elevation given here is the depth in metres from the given vehicle offset to the seabed. An adjustment is made from the position of the altimeter (echo sounder) to the vehicle offset by taking account of pitch and roll of the vehicle. The seabed is assumed to be flat as the given offset could be a short distance away from the altimeter which will mean that the depth given here is not truly accurate.

A.6.1.306 SP2.Offsets.Pos1.Lat

[Real Number]

Default caption in log files: "SP2 Offset 1 Latitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LAT | UNIT_ANGLE | SCAN_SLOWLOG |

LOG_VALUE

Position (latitude) of SP2 (mobile or ship) offset #1. Offset #1 is the CRP.

A.6.1.307 SP2.Offsets.Pos1.Lon

[Real Number]

Default caption in log files: "SP2 Offset 1 Longitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LON | UNIT_ANGLE | SCAN_SLOWLOG | LOG_VALUE

Position (longitude) of SP2 (mobile or ship) offset #1. Offset #1 is the CRP.

A.6.1.308 SP2.Offsets.WGS84.Pos1.Alt

[Real Number]

Default caption in log files: "SP2 Offset 1 Altitude (WGS84)"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | SCAN_SLOWLOG | LOG_VALUE

Position (altitude) of SP2 offset #1 converted to WGS84. Offset #1 is the CRP.

A.6.1.309 SP2.Offsets.WGS84.Pos1.Lat

[Real Number]

Default caption in log files: "SP2 Offset 1 Latitude (WGS84)"
Attribute flags: TYPE_DOUBLE | TYPE2_LAT | UNIT_ANGLE | SCAN_SLOWLOG | LOG_VALUE

Position (latitude) of SP2 converted to WGS84 (mobile or ship) offset #1. Offset #1 is the CRP.

A.6.1.310 SP2.Offsets.WGS84.Pos1.Lon

[Real Number]

Default caption in log files: "SP2 Offset 1 Longitude (WGS84)"
Attribute flags: TYPE_DOUBLE | TYPE2_LON | UNIT_ANGLE | SCAN_NOLOG |

LOG_VALUE

Position (longitude) of SP2 converted to WGS84 (mobile or ship) offset #1. Offset #1 is the CRP.

A.6.1.311 SP2.Pos.Alt

[Real Number]

Default caption in log files: "SP2 Altitude"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The position of SP2 (altitude). The position given here is in your selected working datum / vertical reference which is not necessarily WGS84.

A.6.1.312 SP2.Pos.Elev

[Real Number]

Default caption in log files: "SP2 Elevation"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The elevation of SP2 above the seabed

A.6.1.313 SP2.Pos.Lat

[Real Number]

Default caption in log files: "SP2 Latitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LAT | UNIT_ANGLE | LOG_VALUE

The geodetic position of SP2 (latitude). The position given here is in your selected working datum which is not necessarily WGS84.

A.6.1.314 SP2.Pos.Lon

[Real Number]

Default caption in log files: "SP2 Longitude"

Attribute flags: TYPE_DOUBLE | TYPE2_LON | UNIT_ANGLE | LOG_VALUE

The geodetic position of SP2 (longitude). The position given here is in your selected working datum which is not necessarily WGS84.

A.6.1.315 SP2.Positioning

[String]

Default caption in log files: "SP2 Positioning"
Attribute flags: TYPE_STRING

The positioning mode of the vehicle which is currently SP2

A.6.1.316 SP2.Relative.DX

[Real Number]

Default caption in log files: "Plough dx"
Default format specifier: "%+.2f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_NOLOG

The delta X offset of SP2 from the vessel CRP

A.6.1.317 SP2.Relative.DY

[Real Number]

Default caption in log files: "Plough dy"
Default format specifier: "%+.2f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_NOLOG

The delta Y offset of SP2 from the vessel CRP

A.6.1.318 SP2.Relative.DZ

[Real Number]

Default caption in log files: "Plough dz"
Default format specifier: "%+.2f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_NOLOG

The delta Z offset of SP2 from the vessel CRP

A.6.1.319 SP2.Route.Arc.DOL

[Real Number]

Default caption in log files: "SP2 Arc DOL"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The route arc DOL value of SP2. This is the perpendicular distance from route line. On the outer corner of an alter course it is the distance from the corner/arc Alter course radii (if present) are used in this calculation.

A.6.1.320 SP2.Route.Arc.KP

[Real Number]

Default caption in log files: "SP2 Arc Kp"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | TYPE2_KP | UNIT_DISTANCE | UNIT2_KM | LOG_VALUE | SCAN_NOLOG

The route arc KP value of SP2. This is the straight line distance along the route. If no route is active this variable value is undefined (NAN/blank). Alter course radii (if present) are used in this calculation.

A.6.1.321 SP2.Route.DOL

[Real Number]

Default caption in log files: "SP2 DCC"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The route DOL value of SP2. This is the perpendicular distance from route line. On the outer corner of an alter course it is the distance from the corner (in which case the KP value will be the KP of the corner). Alter course radii are ignored by this calculation.

A.6.1.322 SP2.Route.Grid.DOL

[Real Number]

Default caption in log files: "SP2 GRID DOL"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | SCAN_NOLOG | LOG_VALUE

The route GRID DOL value of SP2. This is the perpendicular distance from route line. On the outer corner of an alter course it is the distance from the corner (in which case the KP value will be the KP of the corner). Alter course radii are ignored by this calculation.

A.6.1.323 SP2.Route.Grid.KP

[Real Number]

Default caption in log files: "SP2 GRID kp"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | TYPE2_KP | UNIT_DISTANCE | UNIT2_KM | LOG_VALUE | SCAN_NOLOG

The route GRID KP value of SP2. This is the straight line distance along the route. If no route is active this variable value is undefined (NAN/blank). Alter course radii are ignored by this calculation.

A.6.1.324 SP2.Route.KP

[Real Number]

Default caption in log files: "SP2 kp"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | TYPE2_KP | UNIT_DISTANCE | UNIT2_KM | LOG_VALUE | SCAN_NOLOG

The route KP value of SP2. This is the straight line distance along the route. If no route is active this variable value is undefined (NAN/blank). Alter course radii are ignored by this calculation

A.6.1.325 SP2.Route.SeabedSlope

[Real Number]

Default caption in log files: "SP2 Route (survey) Seabed Slope"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | SCAN_NOLOG | LOG_VALUE

The route KP (as surveyed) seabed slope value under SP2. The data here comes from the water depth information in the active route (if present). The value is otherwise undefined

A.6.1.326 SP2.Route.Section.Bearing

[Real Number]

Default caption in log files: "SP2 Route Section Bearing"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | SCAN_SLOWLOG | UNIT2_DEGREES | LOG_VALUE

The true bearing of the current route section adjacent to SP2

A.6.1.327 SP2.Route.TerrainDist

[Real Number]

Default caption in log files: "SP2 Terrain Dist (km)"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | TYPE2_KP | UNIT_DISTANCE | UNIT2_KM | LOG_VALUE | SCAN_NOLOG

The route terrain distance value of SP2. This is similar to the KP value except the undulations in the route are taken into account. The route terrain distances are calculated by taking the XYZ distances between each point (the route should contain water depth information for this to be effective). Each route point is converted from lat, long, altitide in order to calculate the distance along each route section. The terrain distance for SP2 is therefore effectively the distance of SP2 along this undulating route. If no route is active this variable value is undefined (NAN/blank). Alter course radii are ignored by this calculation NOTE that the active route line should not only contain water depth information (which is used to give an altitude value for each point) but the waypoints should ideally be spaced close enough

together such that the curvature of the earth is not of any great significance. This is because the cartesian distance between each waypoint is used to compute the distance. If waypoints are too far apart then the straight line between each may be sufficiently long to dip well below the seabed. This will mean that the terrain distances will be shorter that may have been expected.

A.6.1.328 SP2.Route.WaterDepth

[Real Number]

Default caption in log files: "SP2 Route (survey) Water Depth"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | SCAN_NOLOG | LOG_VALUE

The route KP (as surveyed) water depth value under SP2. The data here comes from the water depth information in the active route (if present). The value is otherwise undefined

A.6.1.329 SP2.Smoothed.CMG

[Real Number]

Default caption in log files: "SP2 CMG"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | SCAN_SLOWLOG | LOG_VALUE

The smoothed SP2 course made good in degrees.

A.6.1.330 SP2.Smoothed.Speed

[Real Number]

Default caption in log files: "SP2 SpeedMS"
Default format specifier: "%.4f"
Attribute flags: TYPE_DOUBLE | UNIT_SPEED | SCAN_UNIT_ALIAS | SCAN_SLOWLOG | LOG_VALUE

The smoothed SP2 speed in metres per second.

A.6.1.331 SP2.Smoothed.SpeedKmh

[Real Number]

Default caption in log files: "SP2 Speed"
Default format specifier: "%.4f"
Attribute flags: TYPE_DOUBLE | UNIT_SPEED | UNIT2_KM_H | SCAN_SLOWLOG | LOG_VALUE

The smoothed SP2 speed in kilometres per hour.

A.6.1.332 SP2.SP1Relative.DX

[Real Number]

Default caption in log files: "SP1 to SP2 dx"
Default format specifier: "%+.2f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_NOLOG

The delta X offset of SP2 from the vessel SP

A.6.1.333 SP2.SP1Relative.DY

[Real Number]

Default caption in log files: "SP1 to SP2 dx"
Default format specifier: "%+.2f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_NOLOG

The delta Y offset of SP2 from the vessel SP

A.6.1.334 SP2.SP1Relative.DZ

[Real Number]

Default caption in log files: "SP1 to SP2 dx"
Default format specifier: "%+.2f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_NOLOG

The delta Z offset of SP2 from the vessel SP

A.6.1.335 SP2.Speed

[Real Number]

Default caption in log files: "SP2 Speed"
Default format specifier: "%.4f"
Attribute flags: TYPE_DOUBLE | UNIT_SPEED | UNIT2_KM_H | SCAN_SLOWLOG | LOG_VALUE

The SP2 speed in kilometres per hour.

A.6.1.336 SP2.SpeedKmh

[Real Number]

Default caption in log files: "SP2 Speed"
Default format specifier: "%.4f"
Attribute flags: TYPE_DOUBLE | UNIT_SPEED | UNIT2_KM_H | SCAN_ALIAS | LOG_VALUE

The SP2 speed in kilometres per hour.

A.6.1.337 SP2.SpeedMS

[Real Number]

Default caption in log files: "SP2 SpeedMS"
Default format specifier: "%.4f"
Attribute flags: TYPE_DOUBLE | UNIT_SPEED | SCAN_UNIT_ALIAS | LOG_VALUE

The SP2 speed in metres per second.

A.6.1.338 SP2.Target1.Bearing

[Real Number]

Default caption in log files: "SP2 Target1 Bearing"
Default format specifier: "%.1f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | LOG_VALUE

The true bearing in degrees from SP2 to the auxilliary target #1

A.6.1.339 SP2.Target1.Range

[Real Number]

Default caption in log files: "SP2 Target1 Range"
Default format specifier: "%.1f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The range in metres from SP2 to the auxilliary target #1

A.6.1.340 SP2.WaterDepth

[Real Number]

Default caption in log files: "SP2 Water Depth"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The water depth at the SP2 offset position (if SP2 is a mobile)

A.6.1.341 SP2.WGS84.Pos.Alt

[Real Number]

Default caption in log files: "SP2 Altitude (WGS84)"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The position of SP2 (altitude) in WGS84.

A.6.1.342 SP2.WGS84.Pos.Lat

[Real Number]

Default caption in log files: "SP2 Latitude (WGS84)"
Attribute flags: TYPE_DOUBLE | TYPE2_LAT | UNIT_ANGLE | LOG_VALUE

The geodetic position of SP2 (latitude). The position given here is in WGS84.

A.6.1.343 SP2.WGS84.Pos.Lon

[Real Number]

Default caption in log files: "SP2 Longitude (WGS84)"
Attribute flags: TYPE_DOUBLE | TYPE2_LON | UNIT_ANGLE | LOG_VALUE

The geodetic position of SP2 (longitude). The position given here is in WGS84.

A.6.1.344 SP3.Grid.Easting

[Real Number]

Default caption in log files: "SP3 Easting"
Default format specifier: "%+.2f"
Attribute flags: TYPE_DOUBLE | TYPE2_EAST | SCAN_NOLOG | LOG_VALUE

The grid position of SP3 (easting). The position given here is according to your selected map projection and datum.

A.6.1.345 SP3.Grid.Northing

[Real Number]

Default caption in log files: "SP3 Northing"
Default format specifier: "%+.2f"
Attribute flags: TYPE_DOUBLE | TYPE2_NORTH | SCAN_NOLOG | LOG_VALUE

The grid position of SP3 (northing). The position given here is according to your selected map projection and datum.

A.6.1.346 SP3.GridHeading

[Real Number]

Default caption in log files: "SP3 Grid Heading"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | LOG_VALUE

Grid heading based on delta easting/northing and not a true direction. This is computed using mobile heading obtained the vehicle gyro or other source unless SP2 is an offset on the ship in which case the ship heading is used.

A.6.1.347 SP3.Heading

[Real Number]

Default caption in log files: "SP3 Heading"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | LOG_VALUE

The true heading for SP3.

A.6.1.348 SP3.LaybackBearing

[Real Number]

Default caption in log files: "SP3 Layback Bearing"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | LOG_VALUE | SCAN_NOLOG

The layback bearing of SP3 from the Ship.LaybackPoint

A.6.1.349 SP3.LaybackDistance

[Real Number]

Default caption in log files: "SP3 Layback Distance"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_NOLOG

The layback distance of SP3 from the Ship.LaybackPoint

A.6.1.350 SP3.LaybackMode

[String]

Default caption in log files: "SP3 LaybackMode"
Attribute flags: TYPE_STRING

The positioning mode of the vehicle which is currently SP3

A.6.1.351 SP3.Motion.Pitch

[Real Number]

Default caption in log files: "SP3 Pitch"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | LOG_VALUE

The pitch of SP3 (in degrees) as obtained from a motion sensor. If SP3 is a vessel offset then this variable has the same value as Ship.Motion.Pitch otherwise if SP3 is a mobile the variable has the value of the mobile pitch or zero if this is not known.

A.6.1.352 SP3.Motion.Roll

[Real Number]

Default caption in log files: "SP3 Roll"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | LOG_VALUE

The roll of SP3 (in degrees) as obtained from a motion sensor. If SP3 is a vessel offset then this variable has the same value as Ship.Motion.Roll otherwise if SP3 is a mobile the variable has the value of the mobile roll or zero if this is not known.

A.6.1.353 SP3.Name

[String]

Default caption in log files: "SP3 Name"
Attribute flags: TYPE_STRING

The name of the vehicle (or vessel offset) which is currently SP3

A.6.1.354 SP3.Offset.Pos.Name

[String]

Default caption in log files: "SP3 Pos Offset Name"
Attribute flags: TYPE_STRING

The name of the vehicle offset which is the positioning offset for SP3 (blank if a vessel offset is used)

A.6.1.355 SP3.Offset.SP.Name

[String]

Default caption in log files: "SP3 SP Offset Name"
Attribute flags: TYPE_STRING

The name of the vehicle offset which is the steer point offset for SP3 (blank if a vessel offset is used)

A.6.1.356 SP3.Offsets.Grid1.Easting

[Real Number]

Default caption in log files: "SP3 Offset 1 Easting"
Default format specifier: "%+.3f"
Attribute flags: TYPE_DOUBLE | TYPE2_EAST | SCAN_NOLOG | LOG_VALUE

Position (easting) of SP3 offset #1. Offset #1 is the CRP.

A.6.1.357 SP3.Offsets.Grid1.Northing

[Real Number]

Default caption in log files: "SP3 Offset 1 Northing"
Default format specifier: "%+.3f"
Attribute flags: TYPE_DOUBLE | TYPE2_NORTH | SCAN_NOLOG | LOG_VALUE

Position (northing) of SP3 offset #1. Offset #1 is the CRP.

A.6.1.358 SP3.Offsets.Pos1.Alt

[Real Number]

Default caption in log files: "SP3 Offset 1 Altitude"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | SCAN_SLOWLOG | LOG_VALUE

Position (altitude) of SP3 (mobile or ship) offset #1. Offset #1 is the CRP. The altitude given here is in your selected working datum (vertical reference) which is not necessarily WGS84.

A.6.1.359 SP3.Offsets.Pos1.Elev

[Real Number]

Default caption in log files: "SP3 Offset 1 Elevation"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | SCAN_SLOWLOG | LOG_VALUE

Elevation from seabed to mobile/vessel offset. Offset #1 is the CRP. The elevation given here is the depth in metres from the given vehicle offset to the seabed. An adjustment is made from the position of the altimeter (echo sounder) to the vehicle offset by taking account of pitch and roll of the vehicle. The seabed is assumed to be flat as the given offset could be a short distance away from the altimeter which will mean that the depth given here is not truly accurate.

A.6.1.360 SP3.Offsets.Pos1.Lat

[Real Number]

Default caption in log files: "SP3 Offset 1 Latitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LAT | UNIT_ANGLE | SCAN_SLOWLOG | LOG_VALUE

Position (latitude) of SP3 (mobile or ship) offset #1. Offset #1 is the CRP.

A.6.1.361 SP3.Offsets.Pos1.Lon

[Real Number]

Default caption in log files: "SP3 Offset 1 Longitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LON | UNIT_ANGLE | SCAN_SLOWLOG | LOG_VALUE

Position (longitude) of SP3 (mobile or ship) offset #1. Offset #1 is the CRP.

A.6.1.362 SP3.Offsets.WGS84.Pos1.Alt

[Real Number]

Default caption in log files: "SP3 Offset 1 Altitude (WGS84)"
Default format specifier: "%.3f"

Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | SCAN_SLOWLOG | LOG_VALUE

Position (altitude) of SP3 offset #1 converted to WGS84. Offset #1 is the CRP

A.6.1.363 SP3.Offsets.WGS84.Pos1.Lat

[Real Number]

Default caption in log files: "SP3 Offset 1 Latitude (WGS84)"
Attribute flags: TYPE_DOUBLE | TYPE2_LAT | UNIT_ANGLE | SCAN_SLOWLOG | LOG_VALUE

Position (latitude) of SP3 converted to WGS84 (mobile or ship) offset #1. Offset #1 is the CRP.

A.6.1.364 SP3.Offsets.WGS84.Pos1.Lon

[Real Number]

Default caption in log files: "SP3 Offset 1 Longitude (WGS84)"
Attribute flags: TYPE_DOUBLE | TYPE2_LON | UNIT_ANGLE | SCAN_NOLOG | LOG_VALUE

Position (longitude) of SP3 converted to WGS84 (mobile or ship) offset #1. Offset #1 is the CRP.

A.6.1.365 SP3.Pos.Alt

[Real Number]

Default caption in log files: "SP3 Altitude"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The position of SP2 (altitude). The position given here is in your selected working datum / vertical reference which is not necessarily WGS84.

A.6.1.366 SP3.Pos.Elev

[Real Number]

Default caption in log files: "SP3 Elevation"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The elevation of SP3 above the seabed

A.6.1.367 SP3.Pos.Lat

[Real Number]

Default caption in log files: "SP3 Latitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LAT | UNIT_ANGLE | LOG_VALUE

The geodetic position of SP3 (latitude). The position given here is in your selected working datum which is not necessarily WGS84.

A.6.1.368 SP3.Pos.Lon

[Real Number]

Default caption in log files: "SP3 Longitude"
Attribute flags: TYPE_DOUBLE | TYPE2_LON | UNIT_ANGLE | LOG_VALUE

The geodetic position of SP3 (longitude). The position given here is in your selected working datum which is not necessarily WGS84.

A.6.1.369 SP3.Positioning

[String]

Default caption in log files: "SP3 Positioning"
Attribute flags: TYPE_STRING

The positioning mode of the vehicle which is currently SP3

A.6.1.370 SP3.Relative.DX

[Real Number]

Default caption in log files: "SP3 dx"
Default format specifier: "%+.2f"

Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_NOLOG

The delta X offset of SP3 from the vessel CRP

A.6.1.371 SP3.Relative.DY

[Real Number]

Default caption in log files: "SP3 dy"
Default format specifier: "%+.2f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_NOLOG

The delta Y offset of SP3 from the vessel CRP

A.6.1.372 SP3.Relative.DZ

[Real Number]

Default caption in log files: "SP3 dz"
Default format specifier: "%+.2f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_NOLOG

The delta Z offset of SP3 from the vessel CRP

A.6.1.373 SP3.Route.Arc.DOL

[Real Number]

Default caption in log files: "SP3 Arc DOL"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The route arc DOL value of SP3. This is the perpendicular distance from route line. On the outer corner of an alter course it is the distance from the corner/arc Alter course radii (if present) are used in this calculation.

A.6.1.374 SP3.Route.Arc.KP

[Real Number]

Default caption in log files: "SP3 Arc Kp"

Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | TYPE2_KP | UNIT_DISTANCE | UNIT2_KM | LOG_VALUE | SCAN_NOLOG

The route arc KP value of SP3. This is the straight line distance along the route. If no route is active this variable value is undefined (NAN/blank).Alter course radii (if present) are used in this calculation.

A.6.1.375 SP3.Route.DOL

[Real Number]

Default caption in log files: "SP3 DCC"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | SCAN_NOLOG | LOG_VALUE

The route DOL value of SP3. This is the perpendicular distance from route line. On the outer corner of an alter course it is the distance from the corner (in which case the KP value will be the KP of the corner). Alter course radii are ignored by this calculation.

A.6.1.376 SP3.Route.Grid.DOL

[Real Number]

Default caption in log files: "SP3 GRID DOL"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | SCAN_NOLOG | LOG_VALUE

The route GRID DOL value of SP3. This is the perpendicular distance from route line. On the outer corner of an alter course it is the distance from the corner (in which case the KP value will be the KP of the corner). Alter course radii are ignored by this calculation.

A.6.1.377 SP3.Route.Grid.KP

[Real Number]

Default caption in log files: "SP3 GRID kp"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | TYPE2_KP | UNIT_DISTANCE | UNIT2_KM |

LOG_VALUE | SCAN_NOLOG

The route GRID KP value of SP3. This is the straight line distance along the route. If no route is active this variable value is undefined (NAN/blank). Alter course radii are ignored by this calculation.

A.6.1.378 SP3.Route.KP

[Real Number]

Default caption in log files: "SP3 kp"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | TYPE2_KP | UNIT_DISTANCE | UNIT2_KM | LOG_VALUE | SCAN_NOLOG

The route KP value of SP3.This is the straight line distance along the route. If no route is active this variable value is undefined (NAN/blank).Alter course radii are ignored by this calculation

A.6.1.379 SP3.Route.SeabedSlope

[Real Number]

Default caption in log files: "SP3 Route (survey) Seabed Slope"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | SCAN_NOLOG | LOG_VALUE

The route KP (as surveyed) seabed slope value under SP3. The data here comes from the water depth information in the active route (if present). The value is otherwise undefined

A.6.1.380 SP3.Route.Section.Bearing

[Real Number]

Default caption in log files: "SP3 Route Section Bearing"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | SCAN_SLOWLOG | UNIT2_DEGREES | LOG_VALUE

The true bearing of the current route section adjacent to SP3

A.6.1.381 SP3.Route.TerrainDist

[Real Number]

Default caption in log files: "SP3 Terrain Dist (km)"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | TYPE2_KP | UNIT_DISTANCE | UNIT2_KM | LOG_VALUE

The route terrain distance value of SP3. This is similar to the KP value except the undulations in the route are taken into account. The route terrain distances are calculated by taking the XYZ distances between each point (the route should contain water depth information for this to be effective). Each route point is converted from lat, long, altitide in order to calculate the distance along each route section. The terrain distance for SP3 is therefore effectively the distance of SP3 along this undulating route. If no route is active this variable value is undefined (NAN/blank). Alter course radii are ignored by this calculation NOTE that the active route line should not only contain water depth information (which is used to give an altitude value for each point) but the waypoints should ideally be spaced close enough together such that the curvature of the earth is not of any great significance. This is because the cartesian distance between each waypoint is used to compute the distance. If waypoints are too far apart then the straight line between each may be sufficiently long to dip well below the seabed. This will mean that the terrain distances will be shorter that may have been expected.

A.6.1.382 SP3.Route.WaterDepth

[Real Number]

Default caption in log files: "SP3 Route (survey) Water Depth"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | SCAN_NOLOG | LOG_VALUE

The route KP (as surveyed) water depth value under SP3. The data here comes from the water depth information in the active route (if present). The value is otherwise undefined

A.6.1.383 SP3.Smoothed.CMG

[Real Number]

Default caption in log files: "SP3 CMG"
Default format specifier: "%.2f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | SCAN_SLOWLOG | LOG_VALUE

The smoothed SP2 course made good in degrees.

A.6.1.384 SP3.Smoothed.Speed

[Real Number]

Default caption in log files: "SP3 SpeedMS"
Default format specifier: "%.4f"
Attribute flags: TYPE_DOUBLE | UNIT_SPEED | SCAN_UNIT_ALIAS | SCAN_SLOWLOG | LOG_VALUE

The smoothed SP3 speed in metres per second.

A.6.1.385 SP3.Smoothed.SpeedKmh

[Real Number]

Default caption in log files: "SP3 Speed"
Default format specifier: "%.4f"
Attribute flags: TYPE_DOUBLE | UNIT_SPEED | UNIT2_KM_H | SCAN_SLOWLOG | LOG_VALUE

The smoothed SP3 speed in kilometres per hour.

A.6.1.386 SP3.SP1Relative.DX

[Real Number]

Default caption in log files: "SP1 to SP3 dx"
Default format specifier: "%+.2f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_NOLOG

The delta X offset of SP3 from the vessel SP

A.6.1.387 SP3.SP1Relative.DY

[Real Number]

Default caption in log files: "SP1 to SP3 dx"
Default format specifier: "%+.2f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_NOLOG

The delta Y offset of SP3 from the vessel SP

A.6.1.388 SP3.SP1Relative.DZ

[Real Number]

Default caption in log files: "SP1 to SP3 dx"
Default format specifier: "%+.2f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE | SCAN_NOLOG

The delta Z offset of SP2 from the vessel SP

A.6.1.389 SP3.Speed

[Real Number]

Default caption in log files: "SP3 Speed"
Default format specifier: "%.4f"
Attribute flags: TYPE_DOUBLE | UNIT_SPEED | UNIT2_KM_H | SCAN_SLOWLOG | LOG_VALUE

The SP3 speed in kilometres per hour.

A.6.1.390 SP3.SpeedKmh

[Real Number]

Default caption in log files: "SP3 Speed"
Default format specifier: "%.4f"
Attribute flags: TYPE_DOUBLE | UNIT_SPEED | UNIT2_KM_H | SCAN_ALIAS |

LOG_VALUE

The SP4 speed in kilometres per hour.

A.6.1.391 SP3.SpeedMS

[Real Number]

Default caption in log files: "SP3 SpeedMS"
Default format specifier: "%.4f"
Attribute flags: TYPE_DOUBLE | UNIT_SPEED | SCAN_UNIT_ALIAS | LOG_VALUE

The SP3 speed in metres per second.

A.6.1.392 SP3.Target1.Bearing

[Real Number]

Default caption in log files: "SP3 Target1 Bearing"
Default format specifier: "%.1f"
Attribute flags: TYPE_DOUBLE | UNIT_ANGLE | UNIT2_DEGREES | LOG_VALUE
Old style name: SP3Target1Bearing

The true bearing in degrees from SP3 to the auxilliary target #1

A.6.1.393 SP3.Target1.Range

[Real Number]

Default caption in log files: "SP3 Target1 Range"
Default format specifier: "%.1f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The range in metres from SP3 to the auxilliary target #1

A.6.1.394 SP3.WaterDepth

[Real Number]

Default caption in log files: "SP3 Water Depth"
Default format specifier: "%.3f"

Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The water depth at the SP3 offset position (if SP3 is a mobile)

A.6.1.395 SP3.WGS84.Pos.Alt

[Real Number]

Default caption in log files: "SP3 Altitude (WGS84)"
Default format specifier: "%.3f"
Attribute flags: TYPE_DOUBLE | UNIT_DISTANCE | LOG_VALUE

The position of SP3 (altitude) in WGS84

A.6.1.396 SP3.WGS84.Pos.Lat

[Real Number]

Default caption in log files: "SP3 Latitude (WGS84)"
Attribute flags: TYPE_DOUBLE | TYPE2_LAT | UNIT_ANGLE | LOG_VALUE

The geodetic position of SP3 (latitude). The position given here is in WGS84.

A.6.1.397 SP3.WGS84.Pos.Lon

[Real Number]

Default caption in log files: "SP3 Longitude (WGS84)"
Attribute flags: TYPE_DOUBLE | TYPE2_LON | UNIT_ANGLE | LOG_VALUE

The geodetic position of SP3 (longitude). The position given here is in WGS84.

A.6.1.398 System.CommsScannerState

[Integer]

Default caption in log files: "BSPEngine comms scanner status"
Default format specifier: "%u"
Attribute flags: TYPE_LONG | LOG_VALUE

The state of the communications accessibility scanner thread in BSPEngine. This

variable is provided purely for debugging purposes to allow for diagnosis of issues in the field it is of no use for any other purpose.

A.6.1.399 System.CoordinateSystem

[String]

Default caption in log files: "Coordinate System"
Attribute flags: TYPE_STRING | LOG_TEXT

The coordinate system key describing the coordinate setup currently in use.

A.6.1.400 System.Date

[String]

Default caption in log files: "Date"
Attribute flags: TYPE_STRING | TYPE2_DATE | ASSOC_NEXT | SCAN_NOLOG

The system date in DD/MM/YYYY format. See also System.Time

A.6.1.401 System.DBR.CablesRevision

[String]

Default caption in log files: "Clara Cable Database Revision"
Attribute flags: TYPE_STRING | SCAN_SLOWLOG

The Clara database revision. This is a GUID value that changes whenever the cable database is updated.

A.6.1.402 System.DBR.FixfilesRevision

[String]

Default caption in log files: "Ancilliary Fixfiles Database Revision"
Attribute flags: TYPE_STRING | SCAN_SLOWLOG

The overall revision of all project fix xml tree layout files BUT NOT the associated state data files (contents of the !FixLayouts folder but just the .xml files). This is a GUID value that changes whenever the any of these files are updated.

A.6.1.403 System.DBR.FixfullRevision

[String]

Default caption in log files: "Ancilliary Fixfull Database Revision"
Attribute flags: TYPE_STRING | SCAN_SLOWLOG

The overall revision of all project fix xml tree layout files and associated state data files (contents of the !FixLayouts folder). This is a GUID value that changes whenever the any of these files are updated.

A.6.1.404 System.DBR.FixlayoutRevision

[String]

Default caption in log files: "Fixlayout Database Revision"
Attribute flags: TYPE_STRING | SCAN_SLOWLOG

The fix layout file revision (!NavFix.cfg). This is a GUID value that changes whenever the fix layout (database) is updated.

A.6.1.405 System.DBR.GeodeticsRevision

[String]

Default caption in log files: "Geodetic Database Revision"
Attribute flags: TYPE_STRING | SCAN_SLOWLOG

The geodetics database revision (NavGeo.dat). This is a GUID value that changes whenever the geodetic database is updated.

A.6.1.406 System.DBR.MobileShapesRevision

[String]

Default caption in log files: "Mobile Shapes Revision"
Attribute flags: TYPE_STRING | SCAN_SLOWLOG

The mobile shapes revision. This is a GUID value that changes whenever any one of the mobile object shape definitions is changed or if objects are added or removed.

A.6.1.407 System.DBR.MobilesRevision

[String]

Default caption in log files: "Mobiles Database Revision"
Attribute flags: TYPE_STRING | SCAN_SLOWLOG

The mobiles database revision. This is a GUID value that changes whenever the mobiles database is updated.

A.6.1.408 System.DBR.RoutesRevision

[String]

Default caption in log files: "Routes Database Revision"
Attribute flags: TYPE_STRING | SCAN_SLOWLOG

The route database revision. This is a GUID value that changes whenever the route database is updated.

A.6.1.409 System.DBR.RoutesShapesRevision

[String]

Default caption in log files: "Route Shapes Revision"
Attribute flags: TYPE_STRING | SCAN_SLOWLOG

The route shapes revision. This is a GUID value that changes whenever any one of the stationary object shape definitions is changed or if objects are added or removed.

A.6.1.410 System.DBR.ShipRevision

[String]

Default caption in log files: "Ship Definition Revision"
Attribute flags: TYPE_STRING | SCAN_SLOWLOG

The vessel definition (database) revision. This is a GUID value that changes whenever the vessel SDF is updated.

A.6.1.411 System.DBR.VarsRevision

[String]

Default caption in log files: "Variable Table Revision"
Attribute flags: TYPE_STRING | SCAN_SLOWLOG

The system variable table revision. This is a GUID value that changes whenever the variable list is updated due to a reload of the INI file.

A.6.1.412 System.Time

[String]

Default caption in log files: "Time"
Attribute flags: TYPE_STRING | TYPE2_TIME | ASSOC_PREV

The system time in HH:MM:SS.SS format. See also System.Date

A.6.1.413 System.Timestamp

[String]

Default caption in log files: "System Timestamp"
Default format specifier: "*dd/mm/yyyy hh:nn:ss.ss"
Attribute flags: TYPE_STRING | TYPE2_DATETIME | SCAN_NOLOG

System date and time.

A.6.1.414 System.VMUsage

[Integer]

Default caption in log files: "BSPEngine virtual memory usage"
Default format specifier: "%u"
Attribute flags: TYPE_LONG | LOG_VALUE

The number of bytes of virtual memory currently used by BSPEngine. This variable is provided for debugging of BSPEngine especially with respect to potential memory leaks.

A.6.1.415 Target1.Name

[String]

Default caption in log files: "Target1 Name"
Attribute flags: TYPE_STRING

Name of the current auxiliary target #1.

A.7 Variable attributes

Variable attributes can be applied to variables when they are defined in the INI file. The most commonly used attribute for a variable is the ability to give a variable a heading (e.g. caption) for its column in a log file. Variables in BSPEngine are actually complex Javascript objects and have more properties than just the value of the variable.

Variable attributes are defined in the INI file in curly braces after the variable definition

```
MyVariable.Value = SomeOther.VariableValue + 1
{ attributes go here }
```

To give the variable named MyVariable.Value a heading for use in log files we declare it as follows:

```
MyVariable.Value = SomeOther.VariableValue + 1   { heading = “My Value” }
```

You can also define other special properties of variables in a similar way. This includes the ability to give a variables a format specifier to indicate how the value should be formatted when outputting.

All attribute definitions are placed in curly brackets and most are in the format:

{ key1=value key2=value}

For example:

{ heading=”My Heading” format=”%.3lf” }

Gives a variable a heading and a numeric format specifier indicating it should be printed to 3 decimal places.

Of course variables can also be formatted as strings using Javascript functions

A.7.1 heading

The heading specifier gives the variable a caption. The text for the heading should be placed in double quotes.

This attribute can be applied to variables defined in the [Variables] section, and for fields defined for a given [LogFile1], [LogFile2] etc.

Example:

```
heading=”a heading”
```

A.7.2 format

The format specifier makes it possible to alter the formatting of a variable when it is output. It is typically used to specify the number of decimal places to be printed.

Example:

```
format=”%.4lf”
```

A.7.2.1 Numeric formats

Numeric formats are similar to the C (and PHP) language printf format specifiers.

As long as a variable has a value that is a valid number then a numeric format can be used.

Some examples:

- %.3lf *prints the value to 3 decimal places*
- %.2lf *prints the value to 2 decimal places*
- %lg *prints the value in exponential format (default decimal places)*
- %lf *prints the value in default format (default decimal places)*
- %.4lg *prints the value in exponential format (4 decimal places)*
- %d *prints the value as an integer*
- %03d *prints as an integer 3 digits padded with leading zeroes.*
- %x *prints the value as a hexadecimal number (lower case)*
- %08x *prints as an 8 digit hex number with leading zeroes.*
- 0x%08x *same as above but prefixed with 0x*

For more information look up printf on google.
http://www.cplusplus.com/reference/cstdio/printf/

A.7.2.2 Special formats

For certain types of variables latitude, longitude, time and date it is possible to use alternative format specifiers.

A.7.2.2.1 Date and time formats

Date and time variables can be formatted using a format specifier such as:

yyyy-mm-dd

or

hh::nn:ss.sss

A.7.2.2.2 Latitude and longitude formats

Latitude or longitude variables can be formatted using format specifiers such as

DDD mm.mmm H

DDD.ddddddd

HDDD mm ss.ssss

A.7.3 Variable calculation dependencies

By default variables defined in the INI file are automatically recalculated whenever any of the variables used in the expression are changed. In most cases this is an acceptable default behaviour. For expressions involving many variables or conditionally using different input variables this default behaviour may result in an excessive number of recalculations of the output variable. This also means that if the output variable has an associated history object that the number of entries added to the history is increased and this can results calculated statistics being artificially skewed.

Consider a variable that depends on ship position and heading. Ship position and heading typically arrive in different messages at different times and possibly different rates. You may only wish to calculate such a variable when the GPS position is updated. To prevent the gyro heading being used as well you need to override this default behaviour.

To do this add an additional attribute

#calc_after[var1,var2,var3...]

where var1, var2, var3 etc. are replaced with the names of variables you want to trigger the recalculation. You can have one or more variables listed here. The variables used in the expression itself are no longer considered when determining

when the output variable should be recalculated and instead your supplied list effectively says calculate the output variable whenever any of the listed variables changes.

A.8 Built in functions

This section describes the built in functions that are available. You can define additional functions by writing your own scripts.

A.8.1 Standard functions

The following functions may be used in expressions e.g. when defining custom variables in the INI file:

A.8.1.1 abs(x)

Compute absolute value.
Returns the absolute value of x.

Parameters
x
Floating point value.

Return Value
The absolute value of x.

A.8.1.2 acos(x)

Compute arc cosine.
Returns the arccosine of x.

Parameters
x
Floating point value.

Return Value
The arccosine value of x.

A.8.1.3 acosh(x)

Compute hyperbolic arc cosine.
Returns the hyperbolic arccosine of x.

Parameters
x
Floating point value.

Return Value
The hyperbolic arccosine value of x.

A.8.1.4 asin(x)

Compute arc sine.
Returns the arcsine of x.

Parameters
x
Floating point value.

Return Value
The arcsine value of x.

A.8.1.5 asinh(x)

Compute hyperbolic arc sine.
Returns the hyperbolic arcsine of x.

Parameters
x
Floating point value.

Return Value
The hyperbolic arcsine value of x.

A.8.1.6 atan(x)

Compute arc tangent
Returns the arctan of x.

Parameters
x
Floating point value.

Return Value
The arctan value of x.

A.8.1.7 atan2(y,x)

Compute arc tangent with two parameters.
Returns the principal value of the arc tangent of y/x, expressed in radians. To compute the value, the function uses the sign of both arguments to determine the quadrant.

Parameters
y
Floating point value representing an y-coordinate.

x
Floating point value representing an x-coordinate.

If both arguments passed are zero, an error occurs and the result will be infinity.

Return Value
Principal arc tangent of y/x, in the interval [-pi,+pi] radians.

A.8.1.8 atanh(x)

Compute hyperbolic arc tangent
Returns the hyperbolic arctan of x.

Parameters
x
Floating point value.

Return Value
The hyperbolic arctan value of x.

A.8.1.9 bin2hex(s)

Convert (binary) string to hexadecimal string
Returns the hexadecimal encoded string.

Parameters
s
String value.

Return Value
String containing hexadecimal encoded digits.

A.8.1.10 ceil(x)

Round up value
Returns the smallest integral value that is not less than x.

Parameters
x
Floating point value.

Return Value The smallest integral value not less than x.

A.8.1.11 chr(n)

Convert numeric value to ASCII character
Returns converted value as string containing a single character.

Parameters
n
Numeric value (in the range 0 to 255)

Return Value String with one character.

A.8.1.12 cos(x)

Compute cosine
Returns the cosine of x.

Parameters
x
Floating point value.

Return Value
The cosine value of x.

A.8.1.13 cosh(x)

Compute hyperbolic cosine
Returns the hyperbolic cosine of x.

Parameters
x
Floating point value.

Return Value
The hyperbolic cosine value of x.

A.8.1.14 deg_offset(a, b)

Performs modulo 360 addition to add angle values.

Parameters
a
Numeric angle value in the range 0 <= a < 360.

b
Value to add (can be negative)

Return Value
The resulting angle value.

A.8.1.15 exp(x)

Compute exponential function
Returns the base-e exponential function of x, which is the e number raised to the power x.

Parameters
x
Floating point value.

Return Value
Exponential value of x.

A.8.1.16 floor(x)

Round down value
Returns the largest integral value that is not greater than x.

Parameters
x
Floating point value.

Return Value
The largest integral value not greater than x.

A.8.1.17 hexdec(s)

Convert hexadecimal string to integer.

Parameters
s
String containing hexadecimal number.

Return Value
Integer value

A.8.1.18 iif(b,v1,v2)

Conditional evaluation.

Parameters
b
Boolean 0 or 1 (non zero is considered true) this parameter is usually the result of a comparison.

v1
Value to be returned if b is true.

v2
Value to be returned if b is false.

Return Value
Either v1 or v2 depending on the value of b.

A.8.1.19 hex2bin(s)

Convert hexadecimal string to binary/ASCII
Returns the resulting binary string. Note that this can potentially contain non-printable characters.

Parameters
s
String value (binary data).

Return Value
String containing decoded characters.

A.8.1.20 ln(x)

Compute natural logarithm
Returns the natural logarithm of x. The natural logarithm is the base-e logarithm, the inverse of the natural exponential function (exp). For base-10 logarithms, a specific function log exists.

Parameters
x
Floating point value. If the argument is negative, a domain error occurs, and the result will be NAN If it is zero, the function returns -infinity.

Return Value Natural logarithm of x.

A.8.1.21 log(x)

Compute base 10 logarithm
Returns the base 10 logarithm of x.

Parameters
x
Floating point value. If the argument is negative, a domain error occurs, and the result will be NAN If it is zero, the function returns -infinity.

Return Value Base 10 logarithm of x.

A.8.1.22 ord(c)

Get ASCII code of first character in string.
Returns ascii code of a character value.

Parameters
c
String value (if the string is longer than once character the remaining characters are ignored).

Return Value
An integer containing the ASCII code value. Note if the string is empty the returned value is -1.

A.8.1.23 sgn(x)

Obtains the sign of a numeric value

Parameters
x
Numeric value.

Return Value
The value -1, 0 or 1 depending on the whether x is negative, zero, or positive respectively.

A.8.1.24 sin(x)

Compute sine
Returns the sine of x.

Parameters
x
Floating point value.

Return Value
The sine value of x.

A.8.1.25 sinh(x)

Compute hyperbolic sine
Returns the hyperbolic sine of x.

Parameters
x
Floating point value.

Return Value
The hyperbolic sine value of x.

A.8.1.26 sqrt(x)

Compute square root
Returns the square root of x.

Parameters
x
Floating point value.

If the argument is negative, a domain error occurs, a NAN will be returned.

Return Value
Square root of x.

A.8.1.27 strcat(s1,s2)

Concatenate two strings.

Parameters
s1
First string.

s2
Second string.

Return Value
String, s1 concatenated with s2

A.8.1.28 strcmp(s1,s2)

Compare two strings.
This function starts comparing the first character of each string. If they are equal to each other, it continues with the following pairs until the characters differ or until the end of one of the the strings is reached.

Parameters
s1
First string.

s2
Second string.

Return Value
Returns an integral value indicating the relationship between the strings: A zero value indicates that both strings are equal. A value greater than zero indicates that the first character that does not match has a greater value in str1 than in str2; And a value less than zero indicates the opposite.

A.8.1.29 stricmp(s1,s2)

Compare two strings (case insentitive).
This function starts comparing the first character of each string. If they are equal to each other, it continues with the following pairs until the characters differ or until the end of one of the the strings is reached.

Parameters
s1
First string.

s2
Second string.

Return Value
Returns an integral value indicating the relationship between the strings: A zero value indicates that both strings are equal. A value greater than zero indicates that the first character that does not match has a greater value in str1 than in str2; And a value less than zero indicates the opposite.

A.8.1.30 stripos(s1, s2, [index])

Find position of substring (case insenstive version)
Searches s1 for the first occurance of s2.

Parameters
s1
String to be seached

s2
String to search for

index [optional]
position to start seaching (0 is the first character position)

Return Value
Integer, -1 if not found, otherwise the 0 based index of the begining of the string s2 in s1.

A.8.1.31 strlen(s)

Get string length
Returns the length of s.

Parameters
s
String

Return Value
Integer, length of the string.

A.8.1.32 strpos(s1, s2, [index])

Find position of substring
Searches s1 for the first occurance of s2.

Parameters
s1
String to be seached

s2
String to search for

index [optional]
position to start seaching (0 is the first character position)

Return Value
Integer, -1 if not found, otherwise the 0 based index of the begining of the string s2 in s1.

A.8.1.33 strtolower(s)

Convert string to lowercase
Returns a string containing all upper case letters replaced with lower case equivalents.

Parameters
s
String

Return Value
String

A.8.1.34 strtoupper(s)

Convert string to uppercase
Returns a string containing all lower case letters replaced with upper case equivalents.

Parameters
s
String

Return Value
String

A.8.1.35 substr(s, start, [len])

Extracts a part of a string
Returns the portion of string specified by the start and length parameters.

Parameters
start
Zero based index of the start position.

len [optional]
Number of characters to extract. If this parameter is omitted then the remainder of s will be returned.

Return Value
Returns the extracted part of string.

A.8.1.36 tan(x)

Compute tangent
Returns the tan of x.

Parameters
x
Floating point value.

Return Value
The tan value of x.

A.8.1.37 tanh(x)

Compute hyperbolic tangent
Returns the hyperbolic tan of x.

Parameters
x
Floating point value.

Return Value
The hyperbolic tan value of x.

A.8.1.38 value(x)

Converts a string to a number
A string containing spaces between digits or sign or containing comma's can be converted to a well formed number using this function.

Parameters
x
String.

Return Value
Returns the number or blank if not a valid number

A.8.2 Special functions

Special functions can typically only be to variable objects defined in the INI file

A.8.2.1 timestampOf(var)

Obtains the timestamp for a given variable

Parameters
var
Variable.

Return Value
Timestamp (real number)

(Windows FILETIME value converted to seconds)

A.8.2.2 historyOf(var)

Obtains the history object for a given variable

Parameters
var
Variable.

Return Value
History Object

See Variable History Objects

A.8.2.3 flagsOf(var)

Obtains the type code flags for a given variable

Parameters
var
Variable.

Return Value
Long

A.8.2.4 variableUpdated(var)

Determines if the specified variable has been updated during the current execution loop of BSPEngine. This function is intended to be used in scripts that conditionally update variables dependant on the value of at least one other variable.

Parameters
var
Variable.

Return Value
Boolean

True if the variable in question has been updated.

A.8.2.5 freq(hist)

Frequency counter.

freq(var_history [, min_period_hz [, min_magnitude [, crossing]]]);

Example:

```
Ship.RollFrequency = freq(historyOf(Ship.Roll),
5, 0.1);
```

Returns:

frequency in Hz. Zero if data is rejected.

var_history:

History object for a variable. This can be obtained for a variable

by using the historyOf function. The variable must be present in the

VarHistory section of the INI file.

min_period_hz (optional):

The minimum period at which average point crossings are rejected.

crossing (optional):

crossing value at which edges are counted. If omitted or 'null', then

the average value of the signal is used.

min_magnitude (optional):

Rejects data that does not exceed this magnitude. If data is rejected

then the return value will be zero. Use to avoid small vibrations

resuling in a high frequency (e.g. if calculating roll then this could

occur due to engine vibration when the vessel is on glass flat water).

A.9 History Objects

History objects are created by defining them in the [VarHistory] section of the INI file. See A.3.1.14 [VarHistory] (p.195)

You can create one history object per variable. If you ever need more than one history object for a given variable just create another variable assigned from the original.

History objects record history for a given variable and allow certain statistics to be calculated. You can also access the raw recorded data. History objects store the data in memory so there is a limit to the amount of data that can be stored. Storing too much data in a history object can seriously impact performance so should be avoided.

The properties of history objects are detailed in the following sections.

Timestamp values are in seconds and are effectively windows FILETIME values converted to seconds.

A 64-bit value representing the number of 100-nanosecond intervals since January 1, 1601

http://msdn.microsoft.com/en-us/library/windows/desktop/ms724284(v=vs.85).aspx

You can write your own Javascript functions to make use of history objects for more sophisticated analysis if needed.

To obtain the history object for a given variable use the expression:

```
historyOf(var)
```

Where var is the name of the variable with the history you want to access.

A.9.1 Properties

History objects provide the following (read-only) properties:

A.9.1.1 length

This, like a Javascript array, returns the number of elements

A.9.1.2 timeRange

This returns the time range of the recorded data in seconds

A.9.1.3 rateHz

This is the sampling interval or 0 if only updated as triggered by updates to the specified variable.

A.9.1.4 secondsPerSample

Analogous to rate but the interval instead of the frequency. e.g. (1 / rateHz)

A.9.1.5 min

The arithmetic minimum value.

A.9.1.6 max

The arithmetic maximum value.

A.9.1.7 avg

The arithmetic average value

A.9.1.8 avgmod2pi

The average angle value for angles in radians 0 to 2pi. This special average is computed by summing the unit vectors for each angle and then computing the resultant.

A.9.1.9 avgmod360

The average angle value for angles in degrees 0 to 360. This special average is computed by summing the unit vectors for each angle and then computing the resultant.

A.9.2 Accessing elements

Elements can be accessed using Javascript array syntax. The values are returned.

e.g.

historyOf(var)[0] returns the value of the first element.

In addition the following two methods can be used:

- **valueAt**(index)
  Returns the value at the given index
- **timestampAt**(index)
  Returns the timestamp at the given index

A.9.3 Additional functions

The history object also supports a method:

- **lowerBound**(ts)

  This method returns the index prior to or at the given timestamp value.

A.10 Using scripts

Scripts can be defined for a variety of purposes

e.g.

- Defining your own constants.
- Creating your own functions to use in expressions in the INI file
- Updating variables conditionally

Scripts are defined by adding a [ScriptInclude] section to the INI file. See section A.3.1.12 [ScriptIncludes]

In addition to the functions defined in section A.8 Built in functions (p.354) you can also use standard Javascript functions.

A.10.1 Defining your own constants

Self explanatory really but by defining your own constants in a .js file you will actually get slightly faster script execution than by defining them in the INI file. All variable objects defined in the INI file are actually somewhat more heavyweight than regular Javascript variables. You can of course define constants in the INI file but these are really just regular BSPEngine variables that happen to be defined with a fixed value.

A.10.2 Defining your own functions

You can define your own functions (taking arguments) as regular Javascript functions and call them from expressions in the INI file. All expressions in the INI file have to fit on one line but functions in a .js file are not subject to this restriction. If you need to write special functions which require conditional logic then writing them in Javascript is highly recommended.

A.10.3 Updating variables conditionally

In the INI file you can define one variable as being the result of operations on another but its unconditional. Suppose you are receiving a message which contains a value and an item identifier. For example the item identifier is a tank number and the value is the fluid level in the tank. You have 4 tanks and you want to record the level in each. You cannot do this directly in the INI file.

Lets say that we decode the incoming message (it doesn't matter what this mystery message looks like) and we have the required data in InputChannel1.Message1.Field1 and Field2. Field1 is the level value, Field2 is the tank number.

To split this data out into 4 tank level variables we first declare the tank level variables in the INI file

Example:

```
[Variables]
Tank1.Level = _INPUT
Tank2.Level = _INPUT
Tank3.Level = _INPUT
Tank4.Level = _INPUT
```

Here instead of specifying an expression for each variable we use the placeholder **_INPUT** this means that we will be able to assign to these variables from a script.

Next we must create some Javascript code.

<<tank_levels.js>>

```
Server.calcUserVarsStage1.connect(updateTankVars);

function updateTankVars() {
          if (variableUpdated(InputChannel1.Message1.Field1) &&
              variableUpdated(InputChannel1.Message1.Field2)
          ) {
                    var t = Number(InputChannel1.Message1.Field2);
                    if (t > 0 && t <= 4) {
                              --t;
                              Tank[t].Level = InputChannel1.Message1.Field1;
                    }
          }
}
```

The first line in the js code is executed once for initialization. It connects an event called calcUserVarsStage1 to our function. It means that BSPEngine will call our function on every cycle of its main processing loop.

In our function we must check to see if the input variables were updated. This will only be true if the mystery message has been received and spilt into the input fields. It is very important to do this check or we will end up using a lot of processing power and achieve nothing.

Next we extract the tank number. Notice that the input field variables are strings so we must coerce the value to a number. We then check that the tank number is valid and in the range 1 to 4.

After this check we could have used a switch statement but all numbered variables declared in the INI file e.g. Tank1, Tank2, Tank3, Tank4 are also accessible as arrays. We must however use a zero based index. So Tank1.Level can also be accessed as Tank[0].Level. We take advantage of this in our script.

A.11 Reserved Words

The following words cannot be used in variable names either because they are reserved in the Javascript language or because they would conflict with existing objects.

A.11.1 Reserved by Javascript language

(Primary language keywords)

- abstract
- boolean
- break
- byte
- case
- catch
- char
- class
- const
- continue
- debugger
- default
- delete
- do
- double
- else
- int
- interface
- let
- long
- native
- new
- null
- package
- private
- protected
- prototype
- public
- return
- short
- static
- super

- enum
- export
- extends
- false
- final
- finally
- float
- for
- function
- goto
- if
- implements
- import
- in
- instanceof
- switch
- synchronized
- this
- throw
- throws
- transient
- true
- try
- typeof
- var
- void
- volatile
- while
- with
- yield

(Classes that should be reserved)

- Array
- Date
- JavaArray
- JavaClass
- JavaObject
- JavaPackage
- Math
- NaN
- Number
- Object
- String

(Special Functions / properties)

A.11.2 Reserved by BSPEngine

- Vars
- Server

A.11.2.1 Vars

This object actually holds all other variables although they are all available at global scope as well. Vars.GPS1.Pos.Lat is the same as GPS1.Pos.Lat is the same as GPS[0].Pos.Lat.

A.11.2.2 Server

This is an object that can be used in scripts. See A.10.3 Updating variables conditionally.

A.12 AIS Filtering

AIS filtering is an option that can be used to significantly reduce the amount of AIS data needing to be logged. For the raw device input variable PortInput.AIS_01 there is another corresponding variable called PortInput.AIS_01.Filtered. This variable exists regardless of whether filtering has been configured. If filtering has not been configured then this variable just holds the unfiltered raw input data.

See section A.5 Communications Device Names (p.207)

A.12.1 Configuring AIS filtering

To configure AIS filtering you set up a new section in the INI file as follows:

```
[AIS_01]
FilterRadius = 50000
FilterUnknown = true
FilterDeferIdents = true;
FilterInclude = "$AIALR"
```

There are a number of options for the AIS filter:

A.12.1.1 FilterRadius=

This sets up a distance filter with the specified distance in metres. This is the primary means of reducing the amount of data that needs to be logged.

A.12.1.2 FilterUnknown=

If this key is present (and set to true) then any message that is not a known AIS message will be automatically excluded from the output. Most AIS systems will output a number of additional messages that BSPEngine does not recognize or decode. There is often very little need to log these messages. If a particular message needs to be logged it can be added to the FilterInclude list. See below.

A.12.1.3 FilterDeferIdents=

Raw AIS messages come in a variety of types but broadly speaking there are two categories: A) messages that give details about the vessel associated with a particular MMSI. B) messages that tell us about the position of an object with a given MMSI number. The category A messages do not tell us the position of an object but only its name and other

information. If FilterDeferIdents is set to true then category A messages will not be passed through until a category B message for the same MMSI number arrives and the position is within the filter radius. This way we avoid logging data (the category A messages) for objects that are outside the radius. If FilterDeferIdents is missing or not set to true then the position messages (category B) will be filtered but the information messages (category A) will only be filtered once a category B message has arrived.

A.12.1.4 FilterInclude=

This key allows any message with a given name to be passed through the filter. For instance if you want to log AIS system alarm messages then include $AIALR as in the above example. Normally you will not have this key present. If you are only interested in positions of nearby vessels then you do not need to pass through any additional messages.

A.13 Some Worked Examples

A.13.1 Logging of vessel track and roll period

Relevant parts of INI file

```
[Nav1]
MsgName             = $GPGGA
MsgType             =  1,0,6
Time                =  2,0,0
Latitude            =  3,0,0
LatitudeChar        =  4,0,0
Longitude           =  5,0,0
LongitudeChar       =  6,0,0
GpsQuality          =  7,0,0
NumSatellites       =  8,0,0
HorizontalDilution  =  9,0,0
Altitude            = 10,0,0
GeoidalSeparation   = 12,0,0
DGPSAge             = 14,0,0

[Nav2]
MsgName             = $GPVTG
MsgType             = 1,0,6
Heading             = 2,0,0
SpeedKmh            = 8,0,0

[Nav3]
MsgName             = $GPZDA
MsgType             = 1,0,0
Time                = 2,0,0
Day                 = 3,0,0
Month               = 4,0,0
```

```
Year                = 5,0,0

[Gyro1]
MsgName               = $HEHDT
MsgType               = 1,0,6
Heading               = 2,0,0


[VarHistory]
Ship.Motion.Roll = { seconds=120 }

[Variables]
Ship.Motion.Roll = MRU1.Roll
Ship.Motion.RollPeriod = 1.0 /
freq(historyOf(Ship.Motion.Roll))

[LogFile1]
Title                                   = GPS
Type                                    = Standard
BaseFileName                            =
VesselTrack_
DurationInHours                         = 2
RateInSeconds                           = 4
Field1 = System.Date  {heading = "Date"}
Field2 = System.Time  {heading = "Time"}
Field3 = GPS1.Pos.Lat  {heading = "Latitude" }
Field4 = GPS1.Pos.Lon  {heading = "Longitude" }
Field5 = Gyro1.Heading  {heading = "Ship Heading"
format="%.3f"}
Field6 = Ship.Motion.Roll {heading = "Ship Roll"
format="%.3f"}
Field7 = Ship.Motion.RollPeriod {heading = "Roll
Period" format="%.3f"}
```

Sample output

```
Date,Time,Latitude,Longitude,Ship Heading,Ship Roll,Roll Period
```

```
12/10/2012,10:24:10.89,24 15.236628 S,148 46.451858 E,193.300,0.488,10.005
12/10/2012,10:24:14.95,24 15.310595 S,148 46.354297 E,197.100,0.088,10.005
12/10/2012,10:24:18.99,24 15.394518 S,148 46.247395 E,196.500,-0.635,9.994
12/10/2012,10:24:23.04,24 15.459977 S,148 46.160460 E,197.100,0.965,10.001
12/10/2012,10:24:27.09,24 15.535678 S,148 46.064371 E,194.600,-0.953,9.992
12/10/2012,10:24:31.13,24 15.610784 S,148 45.966254 E,196.800,0.617,10.004
12/10/2012,10:24:35.19,24 15.684840 S,148 45.869845 E,194.600,-0.066,9.994
12/10/2012,10:24:39.24,24 15.760531 S,148 45.771607 E,187.500,-0.525,9.994
12/10/2012,10:24:43.29,24 15.835372 S,148 45.674750 E,192.600,0.901,10.005
12/10/2012,10:24:47.33,24 15.909588 S,148 45.575479 E,187.500,-0.989,10.000
12/10/2012,10:24:51.38,24 15.985196 S,148 45.479281 E,179.800,0.711,10.005
```

[Nav1] to [Nav3] sections configure the built in GPS decoder

[Gyro1] section configures the built in gyro decoder.

The [VarHistory] section sets up history recording for the Ship.Motion.Roll variable

The [Variables] section defines this variable and the RollPeriod variable is defined using the freq() function which performs a frequency counter operation on the history of the Ship.Motion.Roll

[LogFile1] section defines the layout and headings for the log file, sets up the logging rate and duration of the log file.

A.13.2 Logging of raw or filtered AIS data

Relevant parts of INI file:

```
.
.
.
[CustomDataOutputFormat1]
Delimiter = ""
Terminator = ""
Field1 = PortInput.AIS_01.Filtered
Trigger = PortInput.AIS_01.Filtered
LogToFile = 2
.
.
.
[LogFile2]
Title                = AIS
Type                 = Output
BaseFileName  = FilteredAIS_
Extension            = txt
MaxLines                  = 65536
```

Sample output:

```
!AIVDM,1,1,,A,35D7EH50000`msFLdjD<Lp@J0000,0*20
!AIVDM,2,1,5,B,53P7oa02=;KQl51SN21LPU@<P4m0Ttr22222221@000005KeI<1PC3Cm,0*4D
```

```
!AIVDM,2,1,6,B,55D7EH2W3Oo0Pw?33:0t<D4r04hE9B222222221S2Pk836?os=QPC3Cm,0*35
!AIVDM,2,2,6,B,E28888888888880,2*56
!AIVDM,1,1,,A,35D7EH5000O`msFLdjD<Lp@J0000,0*20
!AIVDM,1,1,,A,333d341000O`sd`Lc7AM=2nP0000,0*0A
!AIVDM,1,1,,B,14V?fN0000O`pSVLdgWS>F:l20SW,0*58
!AIVDM,1,1,,B,14V?fN0000O`pSVLdgWS>F:l20SW,0*58
!AIVDM,1,1,,B,14V?fN0000O`pSVLdgWS>F:l20SW,0*58
!AIVDM,1,1,,A,33udhF5P00O`oU8Ldg0h0?w00000,0*0E
!AIVDM,1,1,,A,33udhF5P00O`oU8Ldg0h0?w00000,0*0E
!AIVDM,1,1,,B,13P9cvPOh0O`l6`Ldd37rUdv00S;,0*1C
```

To log filtered or raw AIS data you need to configure a port as the AIS input by assigning an port to the AIS_01 device name.

To configure logging of raw input data to be logged as received and without CSV formatting it is easiest to configure a custom output format and specify that this be logged to a particular output file (LogToFile=2)

[LogFile2] is configured as a Type=Output log and the file extension is set to .TXT

Finally we place a limit on the number of lines.

To log the raw AIS input data or any other raw input data in a similar way simply replace PortInput.AIS_01.Filtered with PortInput.AIS_01 or a different device name.

A.13.3 Logging of system alerts

TBD

A.13.4 Decoding and logging more complex messages

TBD

NAVSYSTEMS (IOM) LIMITED

Blue Spider

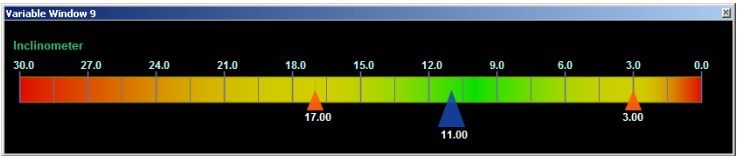
GUIDE TO CUSTOM HTML PANELS

Appendix B

Custom Panels and Watch Windows in Blue Spider

Draft 1.0

B.1 Custom panels and watch windows in Blue Spider



Ordinary watch windows are useful but these are limited to just showing variables and their values. Custom panels (or html plugins) provide a way of displaying richer content and extending the user interface. The gauge shown above was used to help the cable engine operator keep the cable angle between the desired min and max limits. Creating html plugins is not always straightforward and requires a knowledge of html, Javascript, and a good understanding of Blue Spider and BSPEngine. This is something that really should be attempted by the inexperienced! Installing and adapting pre-built plugins however is simple enough.

A plugin is generally just written as a single html file (this may reference others). The main .html file for a plugin is (by convention) placed in the C:\Program Files\NavSystems\Blue Spider\Blue Spider Plugins\NavBed folder. Any .jscript files used by the .html file are placed in C:\Program Files\NavSystems\Blue Spider\Blue Spider Plugins\NavBed\Jscript folder.

When installing a plugin the first thing to do is copy the needed files to these folders. These instructions will be provided with each plugin.

1. Once the files have been placed in the correct folders there are a few simple steps that need to be carried out in Blue Spider.
2. Create a watch window (an empty watch window - don't add any variables)
3. Click right and select Assign Custom Panel...
4. You are presented with a list of already "registered" custom panels.
5. To install a new one select the Install... button and browse to the NavBed folder where you placed the .html file for the plugin you wish to use.
6. This should now appear in the list and you can select it.
7. Pressing the Select button will add the panel to you watch window
8. You will probably need to resize the watch window to a suitable size.

NAVSYSTEMS (IOM) LIMITED

Blue Spider

GUIDE TO GEODETICS

c Appendix C

Geodetics in Blue Spider

Draft 1.0

C.1 Geodetics in Blue Spider

The reader should be familiar with basic geodesy and should have an understanding of datums and vertical reference systems.

C.1.1 Extracts from various internet sources

C.1.1.1 WGS84 - World Geodetic System

The National Geospatial-Intelligence Agency develops, maintains, and enhances the World Geodetic System 1984, the reference frame upon which all geospatial intelligence is based.

The World Geodetic System defines a reference frame for the earth, for use in geodesy and navigation. The latest revision is WGS 84 dating from 1984 (last revised in 2004), which will be valid up to about 2010.

A unified World Geodetic System became essential in the 1950s for several reasons:

- International space science and the beginning of astronautics.
- The lack of inter-continental geodetic information.
- The inability of the large geodetic systems, such as European Datum (ED50), North American Datum (NAD), and Tokyo Datum (TD), to provide a worldwide geo data basis

5 Need for global maps for navigation, aviation, and geography.

C.1.1.2 EGM96 - Earth Gravitational Model 1996

The NASA Goddard Space Flight Centre (GSFC), the National Imagery and Mapping Agency (NIMA), and the Ohio State University (OSU) have collaborated to develop an improved spherical harmonic model of the Earth's gravitational potential to degree 360. The new model, Earth Gravitational Model 1996 (EGM96) incorporates improved surface gravity data, altimeter-derived anomalies from ERS-1 and from the GEOSAT Geodetic Mission (GM), extensive satellite tracking data - including new data from Satellite laser ranging (SLR), the Global Positioning System (GPS), NASA's Tracking and Data Relay Satellite System (TDRSS), the French DORIS system, and the US Navy TRANET Doppler tracking system - as well as direct altimeter ranges from TOPEX/POSEIDON (T/P), ERS-1, and GEOSAT.

The model was used to compute geoid undulations accurate to better than one meter (with the exception of areas void of dense and accurate surface gravity data) and relaize WGS84 as a true three-dimensional reference system. Additional results from the EGM96 solution include models of the dynamic ocean topography to degree 20 from T/P and ERS-1 together, and GEOSAT separately, and improved orbit determination for Earth-orbiting satellites.

C.1.1.3 ETRS89 - Terrestrial Reference System 1989

Most people who are familiar with GPS have heard of the WGS84 (World Geodetic System 1984) coordinate system. This is a global coordinate system designed for use anywhere in the world. WGS84 coordinates are usually expressed as latitude, longitude and ellipsoid height.

WGS84 was designed for navigation applications, where the required accuracy is one metre or lower. A high-accuracy version of WGS84 known as ITRS (International Terrestrial Reference System) has been created in a number of versions since 1989, and this is suitable for international high-accuracy applications (it is used mostly by geoscientists). However, there is a problem with trying to use a global coordinate system for land surveying in a particular country or region. The problem is that the continents are constantly in motion with respect to each other, at rates of up to 12 centimetres per year. There are in reality no fixed points on Earth. In common with the rest of Europe, Great Britain is in motion with respect to the WGS84 coordinate system at a rate of about 2.5 centimetres per year. Over a decade, the WGS84 coordinates of any survey station in Britain change by a quarter of a metre due to this effect, which is unacceptable for precise survey purposes.

For this reason, the European Terrestrial Reference System 1989 (ETRS89) is used as the standard precise GPS coordinate system throughout Europe. ETRS89 is based on ITRS (the

precise version of WGS84), except that it is tied to the European continent, and hence it is steadily moving away from the WGS84 coordinate system. In 2000, the difference between the ITRS (precise WGS84) coordinates of a point and the ETRS89 coordinates is about 25cm, and increasing by about 2.5 cm per year. The relationship between ITRS and ETRS89 is precisely defined at any point in time by a simple transformation published by the International Earth Rotation Service.

The ETRS89 coordinate reference system is used as a standard for precise GPS surveying throughout Europe. Using ETRS89 you can ignore the effects of continental motion: to a high degree of accuracy, the ETRS89 coordinates of a survey station stay fixed, as long as there is no local movement of the survey station. ETRS89 has been officially adopted as a standard coordinate system for precise GPS surveying by most national mapping agencies in Europe, including Ordnance Survey.

C.1.1.4 LAT – Lowest Astronomical Tide

Many national charting agencies, including the United Kingdom Hydrographic Office and the Australian Hydrographic Service, use the Lowest Astronomical Tide (LAT) - the height of the water at the lowest possible theoretical tide - to define chart datum's. LAT is the lowest levels which can be predicted to occur under average meteorological conditions.

Bases are the tide levels for the last 18½ years. The time interval of 18 ½ years considers two oscillation periods substantial for fluctuation of tides: the nutation (wobbly rotation) period of the Moon (18.6 years) in the ecliptic, and their effect, the lunar nodal tidal constituent, is the most important under the longer components of the tide. The values for LAT are computed mathematically by a harmonic analysis.

Advantages

- world-wide uniform map zero for sea charts
- no negative values in tide tables
- secure depth data, the tide can hardly fall further

The advantage of using LAT is that all tidal heights must then be positive (or zero) avoiding possible ambiguity and the need to explicitly state sign. Calculation of the LAT only allows for gravitational effects so lower tides may occur in practice due to other factors (e.g. meteorological effects such as high pressure systems).

C.1.1.5 TAW - Belgian national tidal reference

The Belgian national tidal reference is a reference for height measurements in Belgium and refers to height indications above sea level. All height values in the measurements of the waterways and their valleys and all stages are expressed in accordance with this reference system. The use of this reference system is indicated by the notation m TAW.

C.1.1.6 GEBCO Gridded bathymetry data

The GEBCO data is a terrestrial map of the ocean floor. Its accuracy varies according to location.

GEBCO provides global bathymetry data sets for the world's oceans.

The GEBCO_08 Grid — a global 30 arc-second grid largely generated by combining quality-controlled ship depth soundings with interpolation between sounding points guided by satellite-derived gravity data. However, in areas where they improve on the existing GEBCO_08 grid, data sets generated by other methods have been included.

The GEBCO One Minute Grid — a global one arc-minute grid released: 2003, updated: 2008 — largely based on the most recent set of bathymetric contours contained within the GEBCO Digital Atlas.

C.1.1.7 MSL (Mean Sea Level)

Mean Sea Level is usually defined as being the same as EGM96 although local definitions can also exist.

C.1.2 EGM Geoid Library

C.1.2.1 Introduction

The EGM Geoid Library is an optional extra for use with Blue Spider, BSPEngine and PPT. EGM stands for "Earth Geopotential Model" and EGM96 is a spherical harmonic model of the Earth's gravitational potential complete to degree and order 360. EGM84 is an earlier model with less accuracy and EGM84 and WGS84 are often mistakenly confused with each other and there does not appear to be a standard terminology. This is because WGS84 also refers to a model for MSL but this model is actually called EGM96. Both EGM84 and EGM96 are effectively models of the earth giving the distance between WGS84 ellipsoid height and the geoid height. When we want altitudes to refer to mean sea level rather than the WGS84 ellipsoid we are actually using EGM96. Note that there are other local vertical reference systems such as DVR90.

C.1.2.2 What is a Geoid

The geoid surface is irregular, unlike the reference ellipsoid which is a mathematical idealised representation of the physical Earth, but considerably smoother than Earth's physical surface. Although the physical Earth has excursions

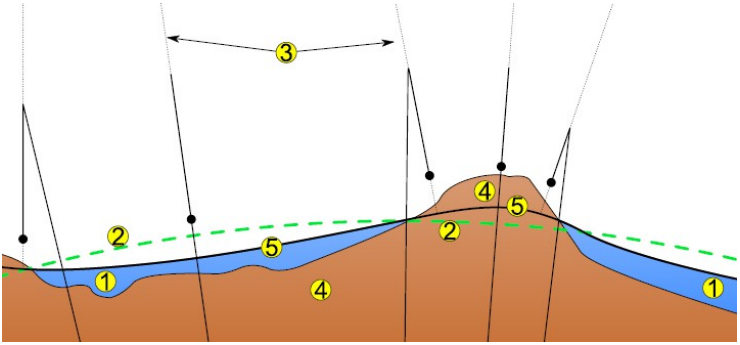
of +8,000 m (Mount Everest) and −11,000 m (Mariana Trench), the geoid's total variation is less than 200 m (-106 to +85 m) compared to a perfect mathematical ellipsoid.

Sea level, if undisturbed by currents and weather, would assume a surface equal to the geoid. If the continental land masses were criss-crossed by a series of tunnels or narrow canals, the sea level in these canals would also coincide with the geoid. In reality the geoid does not have a physical meaning under the continents, but geodesists are able to derive the heights of continental points above this imaginary, yet physically defined, surface by a technique called spirit leveling.

Being an equipotential surface, the geoid is by definition a surface to which the force of gravity is everywhere perpendicular. This means that when travelling by ship, one does not notice the undulations of the geoid; the local vertical is always perpendicular to the geoid and the local horizon tangential component to it. Likewise, spirit levels will always be parallel to the geoid.

Note that a GPS receiver on a ship may, during the course of a long voyage, indicate height variations, even though the ship will always be at sea level (tides not considered). This is because GPS satellites, orbiting about the centre of gravity of the Earth, can only measure heights relative to a geocentric reference ellipsoid. To obtain one's geoidal height, a raw GPS reading must be corrected. Conversely, height determined by spirit levelling from a tidal measurement station, as in

traditional land surveying, will always be geoidal height. Modern GPS receivers have a grid implemented inside where they obtain the geoid (for e.g. EGM-96) height over the WGS ellipsoid from the current position. Then they are able to correct the height above WGS ellipsoid to the height above WGS84 geoid. In that case when the height is not zero on a ship it is because of the tides.



1.	Ocean
2.	Ellipsoid
3.	Local plumb
4.	Continent
5.	Geoid

C.1.2.3 Geoidal Separation

Geoidal Separation is simply the difference between the ellipsoid height and the geoid height. Geoidal separation varies according to location. Most decent GPS receivers output (in the GGA) message an altitude value and a geoidal separation value. The altitude value is typically the altitude with respect to the geoid (EGM96) and the geoidal separation value is the local separation value. The sum of the GPS receivers output values of Altitude and Geoidal separation is effectively the WGS84 ALTITUDE. Note that some GPS receivers do not output a Geoidal separation value at all and in this case the altitude output is WGS84. These are the assumptions Blue Spider makes and if any GPS receiver does not conform to this, then it is probably unsuitable for use.

C.1.2.4 The EGM Geoid Library Install

This is packaged as a separate install (this is because it is quite big and not likely to be frequently updated) - you will probably only have to install it once. With the library installed Blue Spider can work in EGM96 (so can PPT). Using EGM96 is exactly the same as using WGS84 except altitudes are computed to mean sea level. EGM96 only affects altitudes.

To work in EGM96 you simply select EGM96 as the datum instead of WGS84.

In fact there are 3 variations of EGM96:

BL="Bilinear", NS=Natural Spline, SH=Spherical Harmonic, These are all different ways of computing the geoidal separation. In practice the BL method is probably the best to use (EGM96BL). The SH method is included for completeness but is not recommended as it is quite slow! The differences between these techniques (the results computed) are generally very small indeed but in some very specific cases can lead to differences of over 10cm. The BL method is used by most GPS receivers and this is the one generally recommended for use in Blue Spider.

The respective datums you can select are:

EGM96BL, EGM96NS, and EGM96SH

NOTE

The EGM Geoid Library is required in order for the 3D viewer to operate correctly.

C.2 Guide to using vertical datums

C.2.1 Introduction

BSPEngine and Blue Spider provide support for using a wide variety of datums, map projections, and vertical references. The system used internally is underpinned primarily by a 3rd party library 'GeoCalc' which is provided by Blue Marble Geographics. This library is used to perform datum shifts and map projections. The list of supported datums and map projections is defined in a file called geocalc.dat (see also NavGeo.dat which is explained later in this document. The list can be extended to add new ellipsoids, datums and projections by manually editing this configuration file.

Recently it has been necessary to add support for vertical references which are not directly supported by the Blue Marble software. In particular it was necessary to support the DVR90 - Danish Vertical Reference.

To provide the additional functionality required another 3rd party library KMSTrLib2008 is now used.

The support for this library has been added seamlessly in such a way as the user is hardly aware that it is being used. The KMSTRLIB library is executed only if the operator selects a datum that requires its use. The choice of whether to use the library or not is made automatically depending on the chosen working datum. BSPEngine knows whether to use the

library by examining the definition for the selected datum. The datum definitions are defined in the geocalc.dat

(Blue Marble) configuration file. To use DVR90 and other datums supported by KMSTRLIB the geocalc.dat file must be modified (or a modified version supplied) such that the additional datum names are defined. A special tag is added to datum definitions requiring the use of KMSTRLIB. This tells BSPEngine to use addional vertical corrections (via KMSTRLIB.DLL) just for these tagged datums.

Thus the choice of vertical reference is made by selecting the required datum and the datum and vertical reference are effectively combined.

C.2.2 Use of KMSTrLib

KMSTrLib is a transformation library provided by the Danish Ministry of the Environment that we use to perform the vertical datum transformation. BSPEngine and Blue Spider have been extended to support use of this library in order to perform this height transformation. The DLL and data files used by KMSTrLib are included in the latest Blue Spider installation.

In order to allow selection of the correct vertical datum the geocalc.dat file has been updated to include metadata stating that the DVR90 vertical reference should be used (for selected datums).

Blue Spider ascertains via the datum metadata whether DVR90 height adjustment is to be performed.

See http://www.kms.dk/English/Geodesy+and+Surveying/Transformation/

C.2.3 GPS Receiver Datum

BSPEngine expects the raw position and altitude information received from all GPS receivers and other positioning devices to be in the WGS84 datum. If any other input datum should ever need to be used then a software modification will be required.

C.2.4 Datum Shifts

If the operator has selected a datum other than WGS84 then positions (and altitude) from the GPS systems are automatically shifted to this working datum.

C.2.5 Vessel Offsets and Steer Points

The vessel definition defines the relative XYZ position of every possible steer point and also the positions of GPS antenna and other equipment. These offsets are defined relative to the CRP (Common Reference Point). To compute the absolute position (latitude, longitude, altitude) of any offset BSPEngine automatically takes account of the gyro heading and vessel pitch and roll. In this way any vessel offset may be used as a steer point (SP1 or SP2).

C.2.6 Mobile Steer Points

Subsea equipment such as an ROV or Plough can also have a vessel definition defining its own shape and offsets. These offsets can be used as the secondary steer point (SP2). Such equipment is normally positioned using acoustic (HPR) methods (e.g. USBL) but a separate (pressure based) depth sensor is also used. When computing the position of a mobile offset the HPR system gives the (XYZ) position relative to the vessel of a beacon on the mobile. BSPEngine uses the relative position to compute the absolute position by shifting from the vessel reference frame to the mobile beacon position and then to the mobile offset being calculated. In doing this a number of factors are taken into account: Firstly the XYZ position is corrected to be relative to the CRP although this correction is not normally needed as most modern HPR systems are set up such that their CRP coincides with ours. The vessel gyro heading is then used to rotate (in the XY plane) the ship relative heading from the HPR system into a north relative one. This offset is then applied to the already known vessel CRP absolute position in order to determine the position of the beacon on the mobile. Finally the position of the required offset on the mobile is determined by taking into account its relative position and the gyro heading and attitude of the mobile itself.

C.2.7 Subsea depth measurement and adjustment

Subsea depth readings taken by a device that measures water pressure can be corrected to mean sea level or other vertical reference by taking account of the corrected GPS (RTK) altitude and ship heave to measure the actual waterline altitude. In addition the pressure readings can be compensated to take account of the surface pressure (atmosphere) bearing down on the sea.

C.2.8 Depth from pressure sensor or USBL

The depth of a mobile offset can be determined in two ways either by using the depth obtained by the acoustic positioning (HPR) system or by using the pressure based depth sensor. Blue Spider does not attempt to combine these readings but records both and compensates each independently.

WARNING

Using USBL (HPR) for depth determination is not likely to be particularly accurate.

C.2.9 Ship heave sensor

A heave sensor is necessary in order to determine the true waterline and hence correct for Subsea depth measurements. This in turn allows for greater sub-sea positional accuracy. The heave sensor allows us to add a centimetre accuracy offset to our last known GPS altitude

value in order to determine the waterline. The heave sensor is usually part of the motion sensor (pitch & roll) device but can be configured as a separate input.

C.3 Draft

The draft of the vessel is defined as the offset from the waterline to the CRP in calm water. Since the draft of the vessel can change due to changes vessel payload and salinity. The vessel draft is defined in the BSPEngine.INI as an input variable and can be entered by the operator. Both dynamic heave and static draft are actually used to adjust the vessel CRP altitude to the waterline.

C.3.1 Barometric pressure adjustment of Subsea depth readings

The software can be configured to compensate Subsea depth readings made using a pressure sensor - it is assumed that a suitable depth/pressure sensor will be fitted to most subsea vehicles. The Subsea pressure reading is affected by the barometric surface reading. Subsea depth measurements taken in this way will be compensated using the barometer as follows:

Standard atmospheric pressure is 101.325 kPa, pressure increases with water depth at 11.14575 KPa/m therefore the offset (Ship.BarometricDepthAdjustment) to be added is defined as:

```
Ship.BarometricDepthAdjustment = (101.325 -
Ship.Barometer) / 11.14575
```

The actual adjustment method and constants are be fully configurable in BSPEngine.INI.

Using the above compensation method and normal variation in surface pressure the barometric adjustment amount would be typically of the order of +/- 50cm.

C.3.2 Barometer

To perform this barometric adjustment a barometer is required and this is interfaced to BSPEngine as a custom input. The readings from the barometer should be assigned to the variable Ship.Barometer (this variable must be defined in the INI file). The value placed in this (input) variable should be specified in KPa (kilopascals). It is possible, although not recommended (except for test purposes or after failure of the barometer), to define Ship.Barometer as an input variable and have the operator manually enter the barometric pressure.

C.3.3 Configuring the barometer input

In the BSPEngine.INI configuration file a custom input decode section needs to be set up in order to decode the data from the barometer and to activate the device in case it is power cycled. You should define the input decoding in the INI file by adding two sections as shown in the following example:

```
; BAROMETER INPUT (PTB210 Device)
;
[CustomInputFormat6]                              ;
BAROMETER
Field1       = 1,0,0 ; PRESSURE

InputTimeout = After(5) Write(".BP\r")

; If no data is received then after
; a timeout of 5 seconds this
; message is sent to the device to
; turn on the output message.
; After power cycling the
```

```
; barometer it is always necessary.
; to send this command.

[CustomInputChannel6]                              ;
Barometer custom input channel.
Message1 = CustomInputFormat6                      ;
Barometer message
```

In addition the following should be added to the custom log variables [CustomLogVariables] section:

```
Ship.Barometer  =
InputChannel6.Message1.Field1 / 10.0
{#persistent:default=101.325 format="%.4f"
attrs="units:kPa"} ; kPA
```

(note above should all be on a single line)

To instead configure the barometer as a manual input the following should instead be added to the [Variables] section.

```
Ship.Barometer          = _INPUT
{#persistent:default=101.325 format="%.4f"
attrs="units:kPa"} ; kPA
```

(note above should all be on a single line)

Regardless of whether the barometer is configured as a custom or manual input it is also necessary to define the following variable by adding the line below to the [CustomLogVariables] section:

```
; Value to be added to subsea (pressure based)
depth measurements based on barometer
; reading.
; Standard atmospheric pressure is 101.325 kPA
; Pressure increases with water depth at
11.14575 KPa/m
;
Ship.BarometricDepthAdjustment = (101.325 -
Ship.Barometer) / 11.14575 {format="%.2f"
attrs="units:metres"}
```

(note above should all be on a single line)

C.3.4 Draft

To define the ship draft as an input variable add the following to the [Variables] section of BSPEngine.INI:

```
Ship.Draft      = _INPUT
{#persistent:default=0 attrs="units:metres"
format="%.2f"}
```

C.3.5 Echo Sounder

The ships echo sounder position should be defined as a vessel offset using the vessel definition editor. The type for this offset should be set to “Echo Sounder” and the XYZ position determined accurately. The older

'TransducerOffsetToWaterLine' echo sounder configuration in the BSPEngine.INI file should be absent or set to zero.

C.3.6 GPS Receivers

The offsets for these should also be accurately determined and defined in the vessel definition.

C.3.7 HPR System

This should be configured as appropriate for the system being used. It is not normally necessary to define an offset for the HPR pole in fact this should only be done if the system outputs a pole relative rather than CRP relative position. General rule of thumb – don't define a HPR pole offset!

C.3.8 Subsea depth sensors

Vehicle definitions should be defined for mobile vehicles ROV, Plough etc. The position of the depth sensor should be defined as a vehicle offset with the appropriate type.

C.3.9 Motion sensor

Support for most motions sensors used is configured automatically on receipt of data from the sensor. If the motion sensor used supports heave then this is also used. If the sensor does not support heave and a separate heave sensor is used then this must be configured as a custom input and ultimately assign to the variable 'Ship.Motion.Heave'.

C.4 Editing geocalc.dat to define additional datums

Note that a suitable geocalc.dat and NavGeo.dat should have already been supplied prior to the start of operations.

In the geocalc.dat (and in NavGeo.dat) additional datums can be added if required. These are added to the [Datums] section of the file. If KMSTRLIB transformations are required in addition then this is marked by adding a special string to the extended description field. This is always the last but one field. To use KMSTRLIB the extended description field must begin with “#KS:” followed by the target KMSTRLIB datum name. There are a number supported by KMSTRLIB but the following example shows the required configuration to use DVR90 with EUREF89. In the following extract from geocalc.dat two EUREF89 datums are defined. The first is without the DVR90 correction and the second with the correction applied. The string “#KS:dvr90” tells Blue Spider to use the KMSTRLIB DLL to apply a height correction to transform to the DVR90 vertical reference.

```
[Datums]
.
.
.
EUREF89, "EUREF 1989", MOLODENSKY, WGS84, 0, 0, 0, 0, "", 6326
EUREF89-DVR90, "EUREF 1989/DVR90", MOLODENSKY, WGS84, 0, 0, 0, 0,
"#KS:dvr90", 6326
```

You will probably already have these special datums defined in your INI file. However a much simpler type of local vertical datum shift can be defined.

For instance a local vertical datum which has the name GYM and

```
WGS84 GYM, "WGS 1984 GYM", MOLODENSKY, WGS84, 0, 0, 0, 0, "#VShift=-49;#VRef=<ODN>", 6326
```

C.4.1 #Vshift=

This specifies an additional constant vertical shift. Ideal for defining a local vertical datum shift

C.4.2 #Vref=<Name>

This gives the vertical reference a name. Optional but highly recommended as this name will appear in the coordinate system dialog.

NAVSYSTEMS (IOM) LIMITED

Blue Spider

GUIDE TO BONE ANIMATION

Appendix D

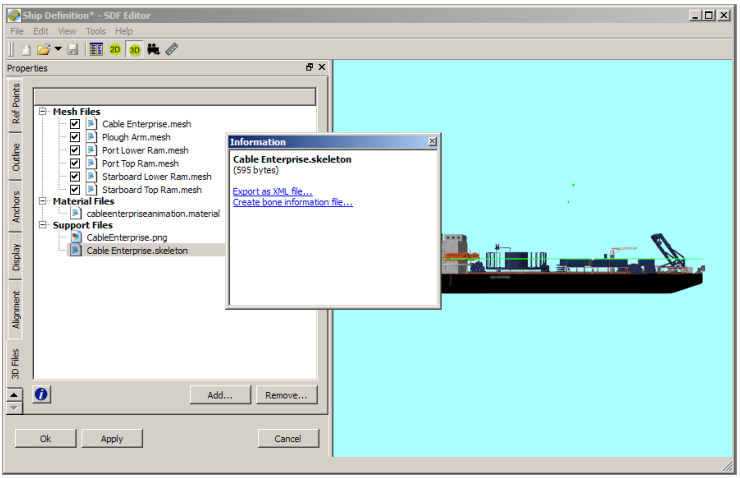
Bone Animation and Scripting

Draft 1.0

D.1 Bone information files

Bone information files allow for direct binding of a BSPEngine variable to control movement of one or more bones in order to animate moving parts of a ship, mobile or even a stationary object. Bone information files can be created for any object and can be created automatically. For more complex animation scripts can be used. A good example of this is animation of the vessel A-frame used for launching a plough. An input variable defined in the INI file will take in some measurement of the arm angle (for instance) and a script will control the position and orientation of each component. In the case of a hydraulic ram the components being the individual pistons. In a complex situation like this it is necessary to define a script to control all of the components based on the angle of the arm. For simpler cases or if you just omit the hydraulic parts you do not need to have a script.

A bone information file can be created and edited in the SdfEditor. Before you can do this you have to already have a mesh (or set of mesh files) which have the necessary bone bindings and also a skeleton file. Once you have the necessary files added to the SDF you can simply select the information button with the skeleton file selected. An option will appear to add the bone information file. You can then open the bone editor which is built in to SdfEdit.



The information button displays summary information for each file. Depending on the file type different options are offered. For a skeleton file if there is no bone information file also present then an option to create one is provided.

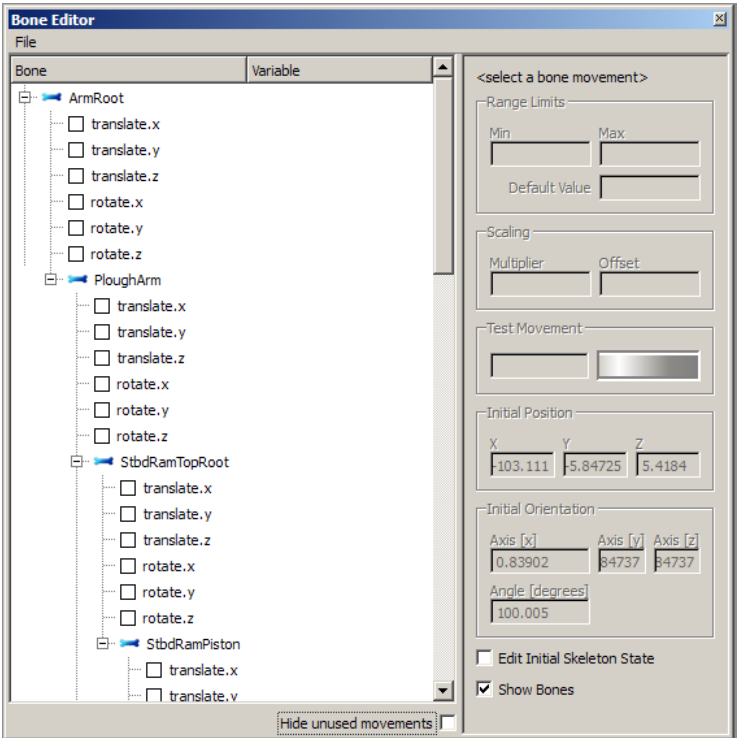
D.1.1 Bone editor

When creating the bone information the bone editor will be automatically displayed. Once this is done the link in the information box changes to **Edit bone information**.

You can open the bone editor by using this link.

The bone editor displays the hierarchical bone structure in a tree view. There will be at least one root bone and these are normally fixed in position. Bones connected to the root bone(s) are shown as child nodes in the tree view. The tree effectively shows how the bones are connected. In reality the bone structure may be more complex. But Ogre skeleton files

can only arrange bones in a simple hierarchical set of one or more trees.

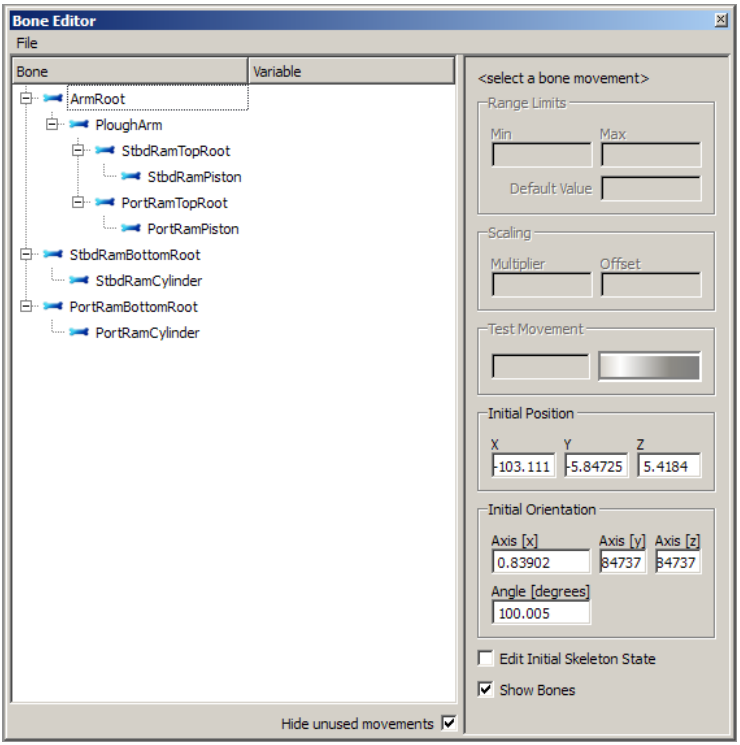


When the bone editor is first opened after creating the bone information for the first time all movements for each degree of freedom of every bone will be displayed.

Each bone has 6 degrees of freedom (DOF). You can try out any movement of any bone by checking the box next to one of the DOF and using the Test Movement wheel. Just to make life interesting the XYZ axis used internally by Ogre and also

in the bone editor is NOT the same as the XYZ frame convention used for a ship, mobile or stationary object.

If you select **Hide unused movements** you can see the bone hierarchy more easily.



This option hides the DOF check boxes that are not checked.

For any DOF you can enter a BSPEngine variable name.

This means that the particular DOF for a given bone will be driven by that variable (unless a script overrides this behaviour).

- **translate.x**
- **translate.y**
- **translate.z**
  These variable values must be in metres

- **rotate.x**
- **rotate.y**
- **rotate.z**
  These variable values must be in degrees

D.2 Bone scripts

Bone scripts can be used to control the positions and orientations of individual bones. The following script is used to animate a vessel A-frame.

```
var arm_angle = new ServerVar('ship.ploughArmAngle');
var plough_arm = new Bone('PloughArm');
var arm_root = new Bone('ArmRoot');
var stbd_ram_piston = new Bone('StbdRamTopRoot');
var port_ram_piston = new Bone('PortRamTopRoot');
var port_ram_cyl = new Bone('PortRamBottomRoot');
var stbd_ram_cyl = new Bone('StbdRamBottomRoot');

var stbd_ram_piston_child = new Bone('StbdRamPiston');
var port_ram_piston_child = new Bone('PortRamPiston');
var port_ram_cyl_child = new Bone('PortRamCylinder');
var stbd_ram_cyl_child = new Bone('StbdRamCylinder');

function onUpdate()
{
                stbd_ram_cyl_child.parentLookAt(stbd_ram_piston);
                port_ram_cyl_child.parentLookAt(port_ram_piston);
                stbd_ram_piston_child.parentLookAt(stbd_ram_cyl);
                port_ram_piston_child.parentLookAt(port_ram_cyl);
}

```

NAVSYSTEMS (IOM) LIMITED

Blue Spider

GUIDE TO THE VESSEL SIMULATOR

Appendix E

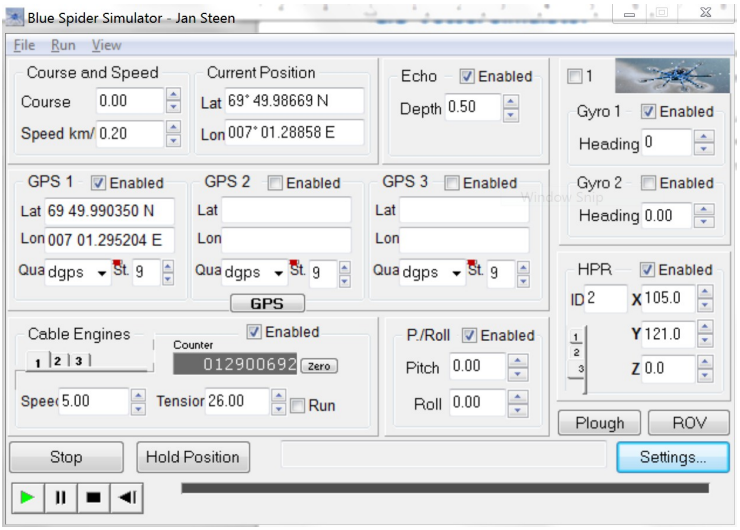
Vessel Simulator

Draft 1.0

E.1 Vessel Simulator

The vessel simulator is of use in testing, demonstration and training or for trying out and evaluating the software.

The vessel simulator can output GPS, Gyro, Echo Sounder, HPR, Motion and some custom inputs. At present the vessel simulator is geared towards the standard NMEA messages but it can also playback captured data.



The main screen of the simulator is designed to take up as little area as possible but allows control over GPS position ,

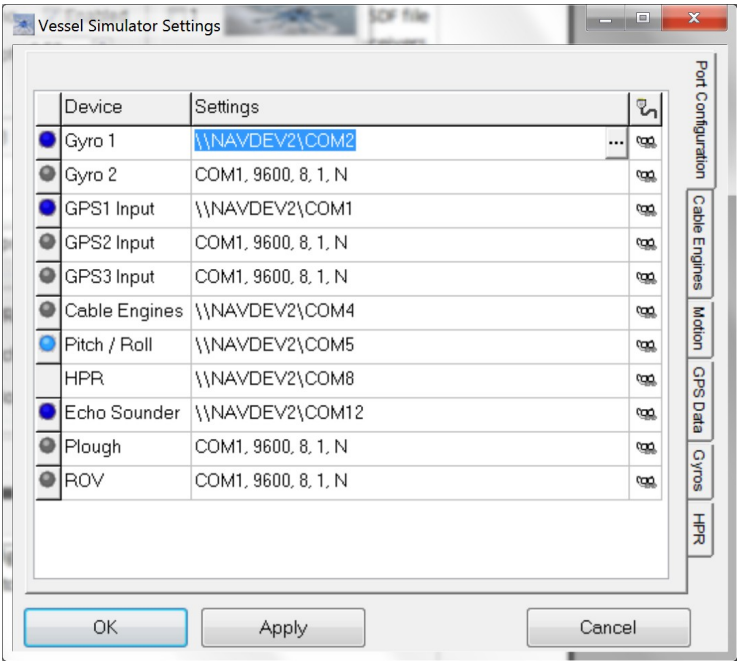
gyro and a few other essentials. For vessels with multiple GPS the vessel simulator can read a vessel definition SDF file and correctly compute the positions of up to 3 gps receivers. In addition it is possible to set up periodic motion for vessel motion such as pitch, roll, heave, surge and sway. The magnitude and timing can be controlled for each. For data from other sources the only option at present is to play back recorded data. In the settings dialog it is possible to not only configure the output ports of the simulator but to also specify the data source. Any simulator output can configured as be the result of playing back data from a text file. The disadvantage is that in playing back data you cant make any user adjustments via the simulator GUI. Future versions of the simulator may allow for script control in order to address this limitation.

Both the simulator and the main Blue Spider software can use virtual COM ports. In testing it is often convenient to use virtual ports as you can then avoid having loads of cables dangling all over the place.

E.2 Simulator Settings

To configure ports in the simulator use the settings dialog and go to the port configuration page.

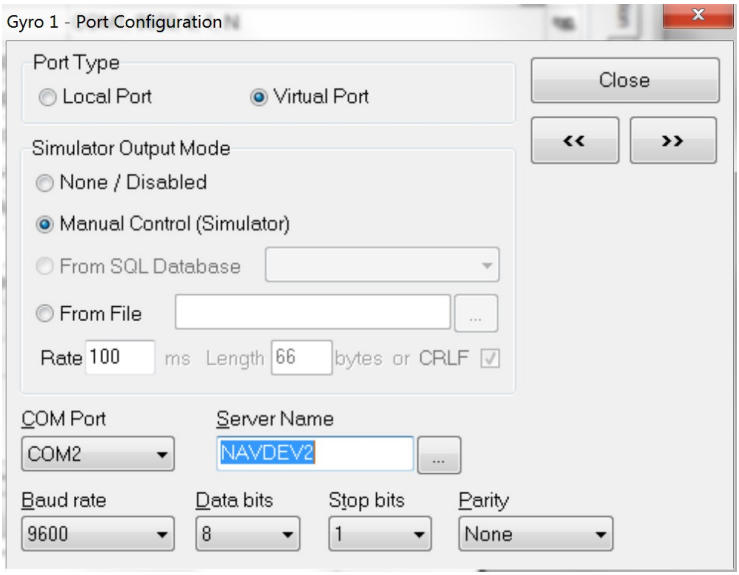
The names of the ports that are present and configurable by the simulator can be changed by loading a port manifest file. This is the PortManifest.cfg file that can be found in the System Config folder. Do this from the main file menu before opening the settings page.



E.2.1 Port Configuration

For each port you can configure by pressing the … button in the Settings column.

This will open the port configuration dialog:



Configure the port as a virtual port and use the same port number in the Blue Spider configuration (and configure as a virtual port there as well)

Alternatively output to a different physical port thats connected to the one configured in Blue Spider.

If you select manual control then output will be under the control of the simulator GUI (providing its an output supported by the simulator).

If you select From File then you can specify the name of the file. For a text file the output is based on a line by line output so if you specify a rate of 100ms you will get ten lines per second. When the end of file is reached it will begin again at the first line.

You can also select from SQL database but this is for playing back multiple ports from the same database. For this option to be available you first have to set up a connection to a SQL server. It is otherwise the same as the file playback option. In order to use the SQL playback you must have previously recorded your data to a SQL database.

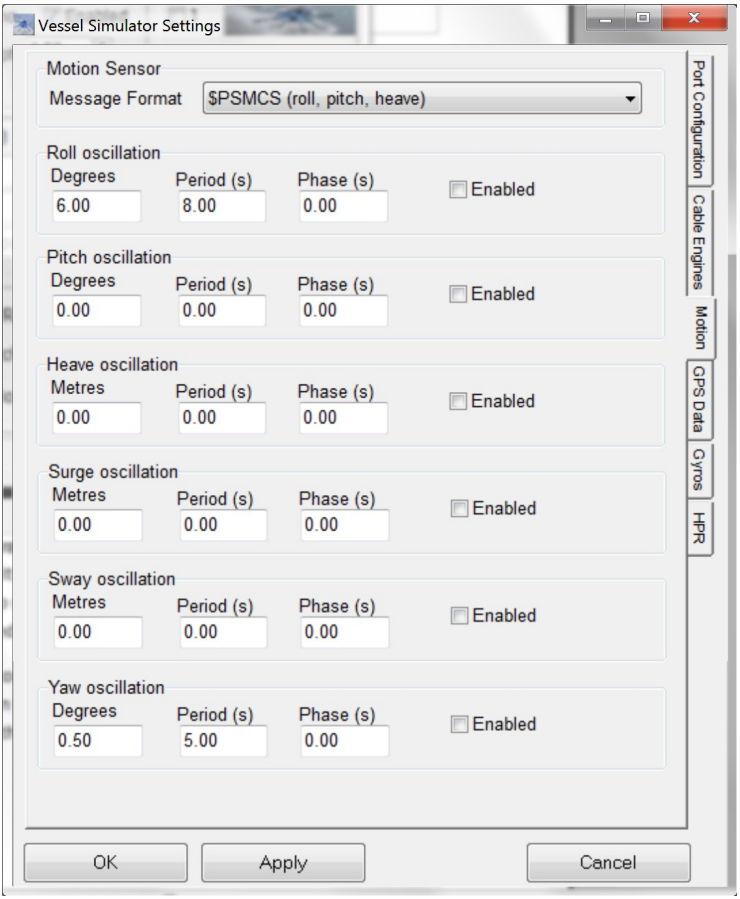
When configuring ports enter your machine name or just a dot (.) in the machine name box unless you want to output via another machine running the BSPNet service.

The arrow buttons << and >> let you view settings for the previous and next ports.

The monitor button lets you view the data being output.

E.2.2 Motion Control

The periodic motion of the ship can be controlled via the Motion page of the Settings dialog.



When motion is enabled the pitch and roll and heave motion output data will vary but in addition for some motions gyro (yaw) and GPS data (all motions) may also be adjusted.